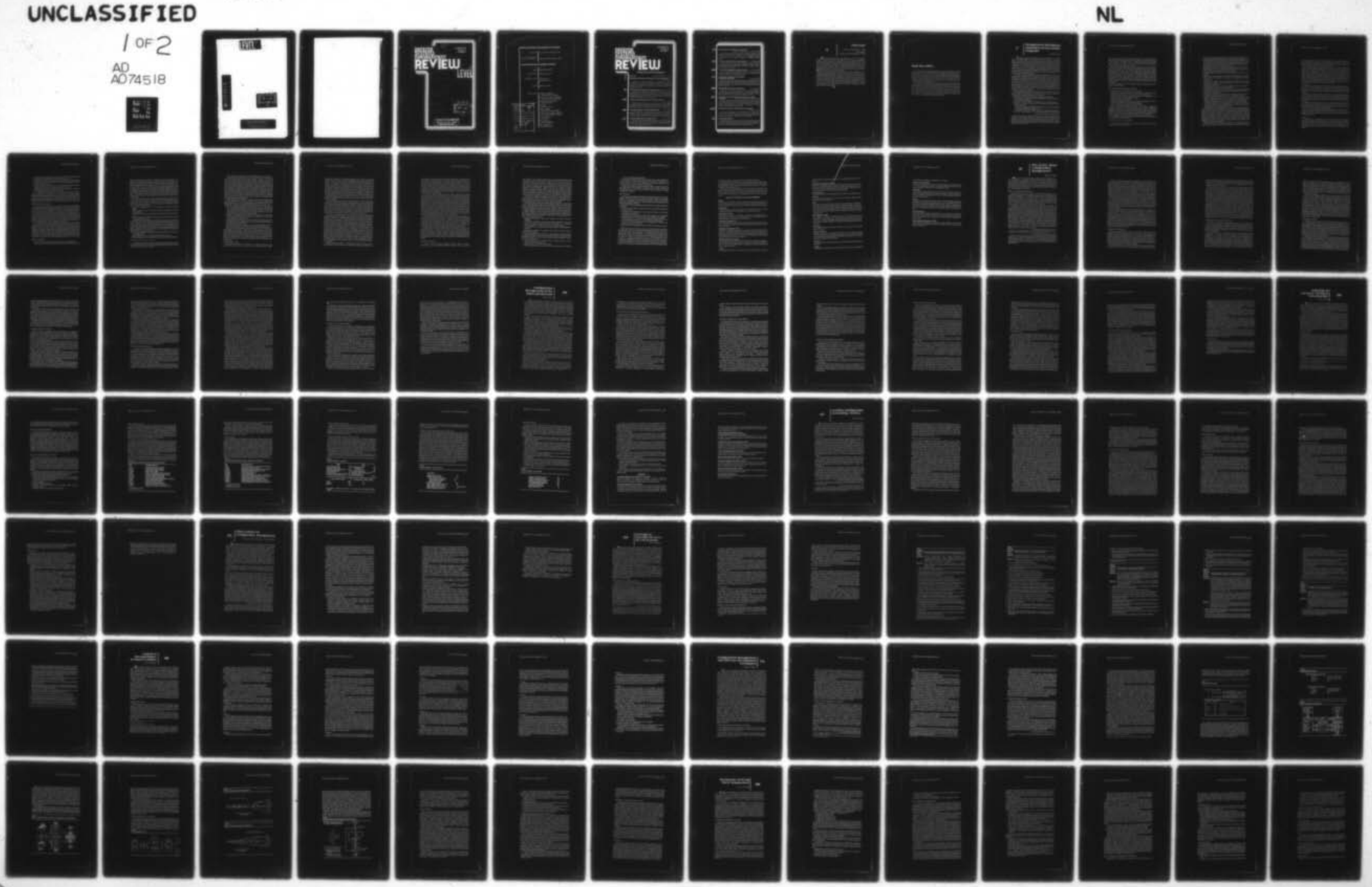
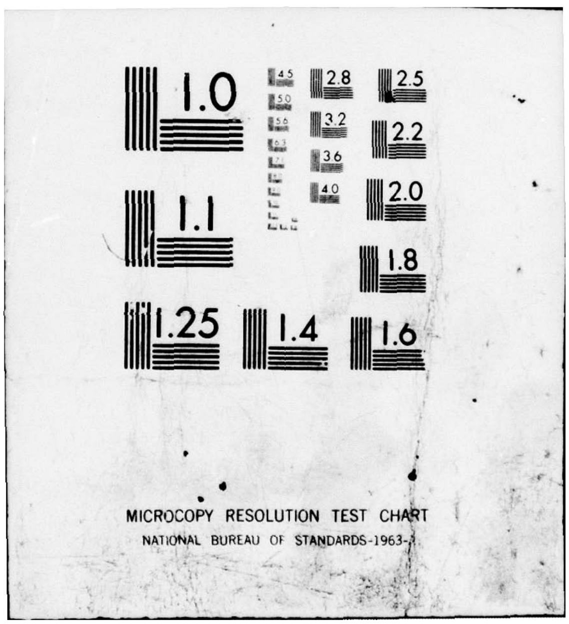
AD-A074 518 DEFENSE SYSTEMS MANAGEMENT COLL FORT BELVOIR VA F/G 5/2
DEFENSE SYSTEMS MANAGEMENT REVIEW. VOLUME 2, NUMBER 3. SUMMER 1--ETC(U)
1979

UNCLASSIFIED NL

1 OF 2
AD
A074518




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL III

AD A074518

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

6

1

DEFENSE SYSTEMS MANAGEMENT REVIEW

SUMMER 1979
VOLUME 2
NUMBER 3

LEVEL III

Volume 2.
Number 3.
Summer 1979.
Configuration and Data Management.

11 1979

12 116p.

410 036

DEFENSE SYSTEMS MANAGEMENT COLLEGE

Commandant	Colonel John B. Hanby, Jr., USA
Dean, Department of Research and Publications	Captain Ronald M. McDivitt, USN

DEFENSE SYSTEMS MANAGEMENT REVIEW

Editor	Robert Wayne Moore
Assistant Editor	Catherine M. Clark
Editorial Assistants	Susan Pollock Wanda R. Jackson
Graphics	Richard B. Brown

SUMMER 1979
VOLUME 2
NUMBER 3

DEFENSE SYSTEMS MANAGEMENT REVIEW

Configuration and Data Management

Foreword

5

Colonel John B. Hanby, Jr., USA
Commandant
Defense Systems Management College

We often talk of acquisition management as if it were a management discipline in and of itself. This is not the case, of course; acquisition management actually involves any number of disciplines and sub-disciplines, each of which has to be considered and successfully carried out before the total program can be called successful. One such discipline is configuration management. The role of the configuration manager is one of the least understood in the acquisition management business. It is the kind of job that no one notices much until there is a failure in carrying out this essential function.

In this issue of the *Review*, we have made an attempt to highlight the configuration management discipline in terms of what it is, how it is done, and some of the problems associated with it. There is no way that we can provide comprehensive coverage of this vital management procedure in one issue, and we lay no claim to having done that. What we do hope to do is increase the awareness as to the true role of the configuration manager among those who depend on their work and, for those who are themselves configuration managers, to provide recommendations and to detail experiences in an effort to make their work both easier and more productive.

from the editor...

We gratefully acknowledge the assistance that the Configuration and Data Management (G-33) Committee of the Electronics Industries Association provided us during the preparation of this issue of the Defense Systems Management Review. *A special thanks is due G-33 Committee Chairman Carl Hershfield and Committee Secretary Albie Strow for their efforts in providing sources for material and then conducting the necessary follow-up. Their guidance and encouragement are greatly appreciated.*

In our next issue we will be addressing some of the major issues/problems that were identified for examination at the Eighth Annual DOD/FAI Acquisition Research Symposium held in May of this year. Some of these issues are relatively new; some have plagued the acquisition process for years. We have selected for the Autumn Review *a few of the more thought-provoking papers addressing these issues that were presented at the symposium. These papers represent the latest thinking on problems that make acquisition management the challenge that it is. Look for the Autumn* DSM Review *late this fall.*

7 Management Disciplines: Harbingers of Successful Programs

David D. Acker

The early program management disciplines came about as a result of a series of initiatives started by government (customer)/industry (contractor) teams. During the 1950s, many of these disciplines evolved without any attempt having been made at higher levels either to control them or to make them uniform throughout the major acquisition programs of the Department of Defense (DOD). Then, in the 1960s, the Office of the Secretary of Defense recognized the need to take a firm stand. Accordingly, it provided specific directions that led to uniform procedures and practices throughout the military services. This arrested the proliferation of management disciplines that were being developed and imposed on major acquisition programs.

The initial program management disciplines implemented by the services tended to delve too deeply into the management prerogatives of contractors. This had the effect of curtailing the development by contractors of new or innovative management approaches. The hew and cry from industry became, "Tell us what you want us to do, not how to do it." This was really a plea for disengagement. It was a plea by industry to DOD to bring about a reduction in the number of stringent management controls and the large number of reporting requirements that military program managers believed necessary to give them management control over contractor efforts. Many of the controls were not appropriate for either fixed-price or incentive contracting.

In the late 1960s a government-industry dialogue was initiated to throw light on this problem. Representatives of the Office of the Secretary of Defense, in discussions with industry representatives, recognized:

- The growing conflicts between program management disciplines;
- The problem of mating the appropriate management disciplines with a particular acquisition program;
- The need to tailor the degree of management control on each program;
- The need to carefully examine each new management discipline before its adoption to ensure:
  - —Its consistency with overall DOD policy and direction and compatibility with other management disciplines; and
  - —Its real need in view of the expense involved in its application and control.

As a result of this dialogue, a new government-industry relationship was

David D. Acker is Professor of Management and Senior Advisor, Defense Systems Management College. Prior to assuming his current position, he was assigned for three years to Plans and Policy, Office of the Director, Defense Research and Engineering, where his responsibilities involved establishing policy and direction in the management disciplines discussed in this article. Mr. Acker spent 23 years in industry in design engineering, project engineering, and program management associated with Air Force, Navy, and Army contracts. He holds B.S. and M.S. degrees in mechanical engineering from Rutgers University.

born. The new relationship permitted the government to manage a program by exception and still react opportunely to each problem on a program requiring a customer solution. This ability to react was made possible through the management information provided at various checkpoints—review and approval points—throughout the acquisition process.

The best management disciplines developed during the 1950s and 1960s have survived the test of time. Those found to be cumbersome have, for the most part, disappeared from the scene. This article deals with three of the program management disciplines that have survived: system engineering management, configuration management, and data management.* While these disciplines may be considered independently, some management disciples consider configuration management and data management as coming under the umbrella of system engineering management. Other disciples treat them separately, but still have difficulty in placing a boundary between configuration management and data management.

This article addresses the three disciplines without entering into the controversy surrounding them.

System Engineering Management Discipline

System engineering is the technical activity used to:

- Transform an operational need into a description of system/end product performance parameters and a system/end product configuration;
- Integrate related technical parameters and ensure compatibility at all physical, functional, and program interfaces;
- Integrate engineering specialties;
- Implement methods for measuring status and taking appropriate action to ensure accomplishment of technical performance;
- Document significant decisions and accomplishments.

System engineering management (SEM) is that program management discipline concerned with (a) the interfacing functions of concern to engineering, manufacturing, materials, logistics, and quality assurance, and (b) the interfacing disciplines of configuration management and data management. The application of SEM by a contractor to any acquisition program must be consistent with the nature, complexity, and scope of the imposed contractual requirements.

In an acquisition program, the system engineering management discipline includes performing the following tasks:

**Author's note:* For concise definitions of these and other terms used herein, refer to the "Glossary of Terms" immediately following this article.

- Planning, controlling, and applying system engineering to transform a contractually defined operational need into a system/end product definition and an optimized design that incorporates equipment, personnel, facilities, computer programs, and procedural data. The definition should be in terms of required system/end product performance parameters and planned technical approaches tailored to the program requirements;
- Identifying, providing, and controlling the detailed definition of the contract work breakdown structure in terms of technical tasks, assuring consistency and correlation of all program technical requirements;
- Identifying high-risk areas and continually assessing their impact on the program;
- Determining program technical requirements and integrating the specialty efforts and such disciplines as configuration management and data management;
- Providing the rationale and the definitive specifications for all hardware/software, facilities, and personnel required to carry out and support contractual requirements;
- Establishing appropriate baselines and management reviews to permit effective engineering change control and monitoring;
- Establishing the rationale for ensuring that engineering decisions leading to the selection of design alternatives are based upon system/end product cost effectiveness considerations;
- Establishing traceability of defined significant engineering decisions to the system engineering management activities on which they are based;
- Planning, evaluating achievement, and reporting technical performance against program objectives for early identification of problems, and for visibility by management so that timely corrective action can be taken;
- Providing appropriate and timely redefinition of program technical requirements in response to changes directed by the customer or the problems identified through evaluation of performance.

The development of a coherent design of a new defense system/end product begins when the nature of the mission to be performed is described and the required uses of the system/end product are established. The process of defining the operational and logistic functions of the system/end product follow. Then the system performance and design requirements are established. Next, detailed design and qualification testing of components takes place. A prototype is built, assembled, and tested and the test data are analyzed and evaluated. After weighing the results of the evaluation against the performance requirements, modifications are made to the design.

Now let's take a closer look at some of the details. The statement of need is expressed in terms of values. These values are based upon consideration of how the

system/end product will be used, the funding that will be available to obtain a given performance, and the date the system/end product will be required. The values are used to establish the standards against which the customer will measure the acceptability of the contractor's design.

The following approach might be used to arrive at the system/end product design. First, the available technology is considered. In one case, off-the-shelf items might be selected for use—either "as is" or with some modification. In another case, completely new items might have to be developed. Second, system/end product performance/effectiveness criteria (i.e., the system/end product capability and availability to perform as and when needed) are considered. Third, a definition of the functional requirements is established. At this point, it is possible to perform trade-off studies—using an iterative process—considering alternative design approaches. The results of these studies will then provide sufficient data from which to select the best design.

APPLICATION

Configuration management and data management are applied throughout the life cycle of a system/end product. Contracts invoking these disciplines usually identify any unique requirements. Normally, the unique requirements are based on the scope of the program and the complexity of the system/end product to be produced.

When properly applied, configuration management and data management are customer-contractor shared responsibilities. Their proper application enhances system/end product performance repeatability, minimizes design change effects, and reduces the incidence of system/end product incompatibility and unusable spare parts.

Configuration management and data management requirements extend beyond the contractor and associate contractors. They extend to subcontractors and suppliers designing and/or fabricating items procured in support of the program on which these disciplines are being applied.

When configuration and data management are applied to a subcontractor's or supplier's privately developed items, the constraints of rights in data need to be recognized as well as the inherent absence of the contractor's right to control the detailed configuration.

TAILORING

The application of configuration and data management must be carefully tailored to be consistent with the quantity, size, scope, stage of life cycle, nature, and complexity of the system/end product involved. Program managers need to tailor the procedures that have been established to the complexity of the system/end product to be managed. When they do, the following results can be expected:

- Earlier and accurate definition, documentation, and tracking of the physical and functional characteristics of the system/end product;
- Availability of verified technical data at the time and for the purpose needed;
- Quicker approval and implementation of proposed engineering changes, waivers, and deviations;
- Increased operational effectiveness of a deployed system/end product, and improved system/end product support at reduced total cost;
- Significant reduction in the number and variety of data, forms, and reports for managing system/end product configuration.

For a complex system/end product (such as a missile, aircraft, or avionics system), configuration management or data management may require application of highly organized procedures to ensure achievement of program objectives. However, for a less complex system/end product (such as a test meter), configuration and data management may require nothing more than control of the applicable specification, followed by acceptance inspection of the units produced.

Establishing a Proper Balance

A logical question at this point is, "How does one go about establishing a proper balance among the management disciplines?" Overcontrol on the part of the customer can be costly and may not allow the contractor sufficient opportunity to be creative in the development of a new system/end product. Undercontrol, on the other hand, may lead to a marginal initial system/end product concept. During the life cycle of a program, this could result in multitudinous design changes to satisfy the customer requirements (objectives) for the system/end product. This means that there must be good communication between the customer and the contractor from the outset of a program—communication that leads to realistic application of the management disciplines.

The establishment of a proper balance is not an easy task. First, after the scope and nature of the management disciplines have been determined, it is often difficult to implement them because, of necessity, they are based upon intangibles. Thus, it is not easy to perceive the difference between a good and a bad management discipline. Added to this is the fact that people can—and do—make a difference in how well the discipline is implemented. Second, the evaluation of a discipline must always be made in relation to values which, by their very nature, are subjective. There are some criteria, principles, and practices that can be applied. When they are applied by knowledgeable individuals, the results are good.

Where It All Begins

Now let's briefly review the acquisition process from the beginning.

The basic policy for managing the acquisition of systems/end products re-

quired by the Department of Defense is defined in DOD Directive 5000.1, Acquisition of Major Defense Systems, originally issued in the summer of 1971. About 30 other DOD directives and instructions relating to the acquisition process have expanded upon the directions set forth in this basic directive. Appropriate sections of the basic directive and supporting documents were revised to comply with the Office of Management and Budget (OMB) Circular No. A-109, Major System Acquisitions, issued by the Office of Federal Procurement Policy (OFPP) in the spring of 1976. Circular A-109, like the basic DOD directive and its supporting documents, recognizes that the acquisition of systems/end products is one of the most crucial and expensive activities conducted to meet our nation's needs.

The OFPP intended Circular A-109 to effect reforms that would prevent cost overruns on future programs and reduce the controversy about whether a new system/end product is needed. The policy set forth by the OFPP is consistent with the 1972 recommendations of the Commission on Government Procurement.[1] This OFPP policy requires:

- Top-level management attention in the determination of mission needs and goals;
- Opportunities for contractors to present innovative approaches to meet the needs and goals;
- Avoidance of premature commitment to full-scale development and production;
- Early communication with the Congress relative to the approach to be taken to meet the needs and goals.

The DOD directives and instructions establish the position of Defense Acquisition Executive. These documents also outline the logical sequence of events in the acquisition process: identify the key decisions to be made during the system/end product life cycle; give the military services some flexibility in carrying out their responsibilities; and advocate that customer program managers be given the authority to trade off performance, cost, and schedules within specified ranges. Furthermore, these documents:

- Place strong emphasis on the initial planning and definition activities in the acquisition process;
- Encourage innovation and conceptual competition by industry;
- Require designation of a customer program manager and establishment of

1. A congressionally established Commission created by Public Law 91-129 to study and recommend methods to promote economy, efficiency, and effectiveness in Federal procurement. Volume 2, Part C, pages 69-187, of the report of the Commission, dated December 1972, contains recommendations relating to the acquisition of major systems.

lines of authority, responsibility, and accountability early in a program.

Program initiation (Milestone Zero) occurs when the Secretary of Defense approves the Mission Element Need Statement (MENS). Passage of this milestone follows reconciliation of the mission need to DOD capabilities, priorities, and resources, and the establishment of program constraints. After MENS approval, a customer program manager is assigned, the acquisition strategy is developed, and a mission-based request for proposal (RFP) is prepared. Following a broad-based competition, preferred alternatives will be selected at Milestone I for trade-off comparison during the demonstration and validation phase of the program. Upon successful completion of the demonstration and validation phase, the Secretary of Defense will approve initiation of full-scale development with possible limited or low-rate production (Milestone II). This phase will be completed when the Secretary of Defense approves quantity production and deployment of the system/end product (Milestone III).

Before continuing, it is important to consider the process of contractor selection. The selection follows customer evaluation of the proposals submitted by competing contractors. To aid in the selection of the best qualified contractor (or contractors), the request for proposal (RFP) prepared by the customer must:

- Provide quantifiable objectives and characteristics such as: key program data; critical system/end product performance parameters; performance, cost, and schedule trade-off limitations; life-cycle cost goals; a list of source selection criteria with an indication of the relative importance of each one; and a limitation on the number of proposal pages;
- Describe non-quantifiable user problems, the anticipated environmental conditions in which the system/end product will operate or be stored, and the anticipated areas of major risk;
- Cause each potential contractor to present a well-conceived plan that clearly demonstrates that his technical solution can be obtained at an affordable cost and within the required schedule;
- Encourage each potential contractor to not only respond directly to the request for proposal, but to offer innovative alternate design approaches.

Upon receipt of proposals from competing contractors, source selection takes place. Ideally, source selection is based upon a realistic appraisal—by an impartial and balanced review team—of the proposals submitted. The final selection is made by the lowest-level DOD authority consistent with the magnitude and importance of the program. Such a decision follows a review of the proposals received to determine which one displays the highest degree of credibility and, at the same time, best meets the customer-established performance objectives at an affordable cost.

The Contractor's Role

Following source selection, the contractor (or contractors) becomes a full partner with the customer in the acquisition process. Through effective system

engineering, configuration, and data management, the customer is guaranteed that the system/end product to be produced and delivered is what he intended it to be, i.e., it is a system/end product that will perform (both physically and functionally) as defined by the specifications and drawings. Furthermore, the customer is assured that the configuration of the system/end product will be identified in sufficient detail so the performance, quality, reliability, and maintenance and support concepts of future systems/end products of the same type can be repeated. The management disciplines that we have highlighted integrate technical and administrative actions by identifying, documenting, controlling, and reporting the physical and functional characteristics of the system/end product as it proceeds through the successive phases of its life cycle. Once the configuration has been adequately identified, modification of the system/end product may be precisely controlled.

The contractor program manager must depend heavily upon his engineering organization to achieve a technically acceptable design of a system/end product that can be easily maintained and readily supported. He must ensure the application of both new and old technology to produce a system/end product that satisfies the customer needs and, at the same time, is capable of economical production, operation, maintenance, and support. Today, everyone concerned with the evolution of a new system/end product must consider cost and schedule objectives as having an importance equal to the performance objectives.

A common system logic needs to be communicated by the contractor program manager and practiced by engineering personnel. The essential characteristics of the logic to which I refer are defining the operational and support requirements; prioritizing the requirements, considering objectively the alternative solutions; and selecting the best solution using sound criteria. The criteria need to be determined on a case-by-case basis considering the following: operational/support effectiveness; the technological, cost, and schedule risks; the simplicity, reliability, and supportability of the design solutions; the benefits of standardization; and the life-cycle costs. Marginal gains in operational performance need to be weighed carefully against possible reductions available in production unit-costs and life-cycle costs. Realistic compromises and trade-offs need to be made to achieve the best possible return on investment. To provide a rational basis for decision-making, the management disciplines used need to be closed-loop and iterative to the extent necessary to continuously relate requirements/objectives (or changes thereto), trade-offs, and solutions. Finally, management needs to focus upon application of logic while, at the same time, keeping documentation to a minimum.

Technical planning should be sufficiently broad and deep to identify: operational, support, and test requirements; development tasks and associated costs

and schedules; technical uncertainty (to enable the phasing of appropriate risk reduction efforts); and achievement milestones. Sufficient flexibility should be incorporated into the development of methods to enable continuous trade-offs to be made between system/end product performance requirements, costs, and scheduled completion dates. Modification or cancellation of the program should be an option if it appears the system/end product being developed will fall short of expectations. An integrated engineering management plan, consolidating the engineering specialty disciplines, should be prepared as an integral part of the basic program management plan.

At the outset, technical planning should consider competition and austere management practices to maximize the return on advanced development investment. Advanced development prototypes need to be used to demonstrate the practicability of one or more of the following: new technology, (which offers significant increases in performance); operational innovations (which should be demonstrated in a representative operational environment); and development, manufacturing, operational, and support concepts (which offer significant cost benefits). The focus at the beginning of a program is on demonstrating the practicability of new technology and operational innovations. However, at that time, some consideration also needs to be given to the total system/end product aspects of the design. By total aspects, I mean the hardware/software, facilities, personnel, and procedural data that drive life-cycle costs and become critical elements in the full-scale engineering development phase of a program.

Development specifications need to be functional rather than detailed and should be in keeping with the objectives of the program. Prioritized requirements and flexible goals need to be identified to enable continuous trade-offs during system/end product development. Care needs to be exercised in the application of specifications and standards, particularly those that are considered to be "boiler plate." A specification tree needs to be structured so as to provide for adequate identification and control with a minimum number of documents.

Program baselines should be established to provide a basis for the control of engineering changes. Care should be exercised in timing the establishment of these baselines so as not to unduly constrain the design or create unnecessary and expensive administrative burdens. Normally, an approved system/end product specification establishes the initial baseline; an approved development specification establishes the next baseline, and a product specification establishes the final baseline.

Assessment of Progress

A good customer program manager assesses technical progress (performance), cost, and schedule throughout the contract. To do so in an effi-

cient manner, he must establish and maintain a close working relationship among the people in his office, the contractor, and the associate contractors, if any. This will help to ensure continuing and meaningful exchange of up-to-date, pertinent data. System/end product users need to participate in progress assessments and refer problems, changing needs, and recommendations to the customer program manager for final disposition. Rapport between the contractor's and the customer's program personnel permits a timely and candid exchange of pertinent information, precludes surprises, and minimizes the impact of program redirections that might follow formal assessments. The scope, frequency, and scheduling of reviews to assess progress must be tailored to meet individual program needs.

The contractor establishes and updates task definitions, resource allocations, and schedules to maintain consistency with initial and revised program requirements/objectives. Also, the contractor reports status periodically—at program milestones established by the customer program manager.

Technical progress is assessed by evaluating engineering results. Documented analyses, trade-off studies, design reviews, hardware simulations, and test results aid in accomplishing this objective. Formal reports are required only when specifically requested by the customer program manager to satisfy a stated need.

Design reviews are conducted to assess the capability of the system/end product under development to meet current customer requirements. Informal communication regarding the evolving design configuration minimizes the length and cost of formal reviews and contributes to early resolution of any problems. Normally, the formal design reviews:

- Address overall progress, major risks, changes to user needs, and major problems;
- Are limited in frequency to preclude unnecessary expenditure of time and funds, yet frequent enough to ensure maintenance of a continuing customer/contractor understanding of current program objectives;
- Are scheduled so as to support major program milestones and management decisions.

Satisfactory completion of each design review means that another program milestone has been passed.

Testing and evaluation are conducted to assess progress at scheduled points throughout the program. The tests are normally conducted as early as practicable to minimize the impact of unknowns and possible program redirection. Special tests are conducted when:

- Technical analyses will not provide sufficiently credible results (e.g., when proof of performance is required);
- Testing is more cost effective than analyses;
- Hardware experience is necessary to make a critical decision (e.g., to choose

one or more alternative approaches).

It should be remembered that test and evaluation are necessary ingredients in the development of a system/end product and they need to be conducted to the extent necessary to provide the performance of hardware/software that is representative of the operational system/end product.

System/end products are usually developed within a specific life cycle cost plan. Although it is not possible to accurately assess costs until completion of development, there is still a need to assess the plan and update it as needed throughout the program.

Final Thoughts

The three major management disciplines discussed in this article are truly harbingers of program success, provided they are properly applied on a program. Therefore, it is imperative that government and industry—the customer and the contractor—program managers become acquainted with them and adapt them to peculiar needs of their programs.

Experience has shown that properly applied program management disciplines will provide:

- Uniformity in the manner in which the customer expresses his requirements to the contractor;
- Uniformly structured responses from the contractor that can be accurately evaluated by the customer;
- For subdivision of large system/end product programs into manageable packages that can be contracted, funded, and controlled incrementally;
- Technical bases for contracts that can adequately support statements of work and thus generate more accurate cost and schedule estimates;
- Controls that are commensurate with the magnitude of financial commitments and the risks involved;
- Controls that can be expanded to match the depth and firmness of information.

In the final analysis, it becomes quite clear that the successful development and deployment of any new system/end product depends upon many factors. Of primary importance is the employment of competent people who are given the responsibility, along with the authority, to accomplish the task. Program objectives must be realistic, but some flexibility to make changes and trade-offs must be granted to program management Risks must be identified early, fully understood, accommodated, and promptly reduced. The customer and contractor program management teams must recognize that each program is different. Further, they must freely communicate knowledge, information, and data that is pertinent to the achievement of program objectives. Finally, the members of these

teams must be highly motivated to achieve the program objectives.

It should be borne in mind that the contract type determines the degree of formality required on a program. A good manager will never invoke more formality than required by contract, unless the anticipated results appear to be worth the expenditure of additional resources.

In the future some changes may be made to the management disciplines with which this article has been concerned. If so, the changes will consist of "fine tuning," rather than alteration of the basic approach because *today's management disciplines will continue to be the harbingers of tomorrow's successful programs.*

Glossary of Principal Terms and Their Definitions

Baseline
An authorized, documented, technical description specifying the physical and functional characteristics of a system/end product. The technical description is used as the basis for configuration control and configuration status accounting.

Configuration
The physical and/or functional characteristics of hardware/software as set forth in technical documentation and achieved in a system/end product.

Configuration Control
The systematic evaluation, coordination, approval, and implementation or disapproval of all changes in the configuration of a system/end product after formal establishment of its configuration identification.

Configuration Identification
The current approved or conditionally approved technical documentation of an item as set forth in specifications, drawings and associated lists, and documents referenced therein.

Configuration Management (CM)
The element of program management that ensures for each program that uniform methods of configuration identification, control, and status accounting are implemented and maintained for a system/end product and that the application of these methods results in effective control of the configuration of a system/end product throughout the life of the program.

Configuration Status Accounting
The recording and reporting of the information that is needed to manage configuration effectively, including a listing of the approved configuration identification, the status of proposed changes to configuration, and the implementation status of approved changes.

Contract
A legal agreement between a customer and a contractor whereby the contractor

commits himself to render specified services or to furnish specified products to the customer.

Contractor and Associate Contractor
Contractor: A company that enters into a contract with an agency of the government to provide specified items of material or supply and/or perform services in accordance with the contractual requirements of the agency. Associate: One of two or more contractors performing on a single program.

Cost Effectiveness
A measure of the value received (effectiveness) for the resources expended (cost).

Customer
A government agency that has executed a contract providing a company with the authority to activate a program.

Data
The means for communication of concepts, plans, descriptions, requirements, and instructions relating to technical projects, material, systems, and services. These may include specifications, standards, engineering drawings, associated lists, manuals, and reports, including scientific and technical reports; they may be in the form of documents, displays, sound recordings, punched cards, or digital or analog data.

Data Management (DM)
That element of program management concerned with the definition and documentation of program data requirements; the assignment of data requirements, schedules, and budgets for data generation; the monitoring of data control systems; and the collection and evaluation of data generation and submittal status.

End Product
That tangible hardware and related software that a contractor has committed himself to produce.

Hardware/Software
Hardware or software, or a combination of both in which the software includes only that associated with hardware in operational use, e.g., computer programs for command and control, handbooks for operations, maintenance, etc. Excludes fabrication specifications and drawings.

Life Cycle
The phases through which a program passes from its initial concept until its final disposition.

Program
The aggregate of controlled events which, from contract award or establishment

to termination, constitutes accomplishment of a contract.

Request for Proposal (RFP)
A document submitted by the customer's procuring agency to potential contractors on a contemplated procurement. It identifies the items to be procured and requests that plans be submitted for accomplishing the effort.

Specification
A document, intended primarily for use in procurement, which clearly and accurately describes the essential and technical requirements for items, materials, or services, including the procedures by which it will be determined that the requirements have been met.

System
A composite of subsystems, assemblies (or sets), skills, and techniques capable of performing and/or supporting an operational (or non-operational) role. A complete system includes related facilities, items, material, services, and personnel required for its operation to the degree that it can be considered a self-sufficient item in its intended operational (or non-operational) and/or support environment.

System Engineering
The process used to define and integrate the use of program and functional resources. The process converts inputs related to customer need into output information that describes that combination of system elements satisfying the customer need.

System Engineering Management (SEM)
The engineering management, direction, and control applied to a system/end product to ascertain and maintain technical integrity over all elements of that system/end product. ‖

21 Why Worry About Configuration Management?

William A. Dean

When your next production item is delivered, will you know what its exact configuration is supposed to be? When the first item is delivered incorporating that new, high-reliability "black box," will the new technical manuals and automatic test equipment be ready to support it? When the contractor completes the qualification test program, can you be sure that he has met all performance requirements? Providing answers to these and similar questions is what configuration management is all about.

What Is Configuration Management?

Configuration management, as defined by the joint DOD regulation on configuration management, is "A discipline applying technical and administrative direction and surveillance to (1) identify and document the functional and physical characteristics of a configuration item; (2) control changes to those characteristics; and (3) record and report change processing and implementation status."* Those are a few very simple words that describe a complex and critical process that must extend over the entire life cycle of a system.

Configuration management requires the careful selection and acquisition of the documentation that describes the unit or system we are buying. The documentation provides the basis for determining whether the unit meets our performance requirements, for establishing a logistics support system for the unit, and for government acceptance of production units. Once the documentation has been placed on contract, configuration management requires that the contractor obtain government agreement before making any change to the documentation. When requesting a change, the contractor must summarize for the government the total impact of the change, especially the impact on the logistics support system. Once the change has been approved, configuration mnagement requires that the government track its implementation to be sure that all new hardware, spares, manuals, etc., are available as proposed. The purpose of configuration management, at the bottom line, is to ensure the continuing logistics supportability of systems in the government inventory.

*AFR 65-3, AR 70-37, NAVMATINST 4130.1A, MCO 4130.1A, DSAR 8250.4, NSA/CSS 80-14, DCAC 100-50,2, and DNA INST 5010.18, *Configuration Management*, 1 July 1974.

William A. Dean is an assistant professor of systems management at the Air Force Institute of Technology's (AFIT) School of Systems and Logistics. He has held a number of positions associated with weapon systems management, culminating in 4 years as a configuration manager with the A-10 Thunderbolt II system program office. He is the creator of the basic configuration management course at AFIT, and has been the course instructor for the past 3 years. Mr. Dean holds a bachelor of engineering degree from Stevens Institute of Technology and a master of science degree from Southern Methodist University.

In trying to understand configuration management, one needs first to understand what is meant by "configuration." The physical aspects of configuration are easy to understand. Everyone can understand configuration when looking at either the physical hardware and the drawings and parts lists that define it, or at the line-by-line listing of the software. On the other hand, few people would consider a set of performance (functional) requirements in a specification to be a configuration. However, early in the development of a new system, the only available description of the system configuration is in terms of performance requirements. Since the physical design will evolve based on these requirements, it is as necessary to control the *functional* configuration throughout the program as it is to control the *physical* configuration developed to meet those requirements.

The basic unit of configuration management is the configuration item (CI). All of the configuration management functions are performed at the CI level. Specifications are written to document the characteristics of each CI; the design reviews and audits are performed for each CI; engineering change proposals are written individually for each CI; and status accounting tracks the implementation of changes to each CI.

The joint DOD regulation defines the configuration item as "an aggregation of hardware/computer programs, or any of their discrete portions, which satisfy an end-use function...." During development and qualification testing, CIs are usually those assemblies of components (such as the guidance unit in an air-to-surface missile) that are critical to successful system performance and that will require separate requirements documentation and careful technical monitoring during design and testing. Once the design has been qualified, additional CIs would include those components (such as the black boxes, the complex printed circuit boards, or a special gyroscope for the guidance unit) that, because they are used for logistic support and have been designated for separate procurement, require separate documentation and control.

Configuration Identification

Configuration identification includes the specifications and their associated diagrams, flow charts, drawings, parts lists, etc., that are used to describe the functional and physical characteristics of a CI. The process of controlling the configuration identification requires that the government establish baselines for various portions of the documentation at appropriate milestones in the program. The exact timing of the establishment of the baselines depends upon the type of program involved, and the configuration manager should have a key input in the timing and in the type of documentation required.

The *functional baseline* is used to document the functional (performance, operational, logistics, training, etc.) requirements for the total system. It usually

consists of a single specification (system specification) describing the essential requirements for the basic functional elements (both hardware and software) of the system. It is usually established at the end of the conceptual phase or during the validation phase of the program.

The *allocated baseline* is used to document the functional requirements for each CI. As such, there may be tens or hundreds of allocated baselines (one for each CI) before an "allocated baseline for the system" (there really is no such thing) exists. The allocated baseline for each CI is documented in a development specification. The requirements in the specification (development specification) are the basis for the contractor's design of the CI; the quality assurance provisions in the specification form the framework for the qualification testing program for the CI. The allocated baseline is usually established at the end of the validation phase or very early in the full-scale engineering development phase of a program.

The *product baseline* is used to document the final physical design that meets the requirements of the allocated baseline for the CI. Product baselines are established for each CI as it successfully completes qualification testing and design/control verification. The specification (product fabrication specification) requirements define the physical design of and the acceptance criteria (performance values) for each production item; the acceptance tests required by the quality assurance section of the product specification must be successfully passed before the government will accept the production item. The product baseline is usually established early in the production phase of the program, but it may be established at or near the end of the full-scale engineering development phase.

Configuration Audits and Design Reviews

The purpose of configuration audits and design reviews is to review the contractor's system engineering (both design and test) efforts while progressive increments of the configuration identification and test documentation are being generated. Although the design reviews are engineering functions, they complement the establishment of the baselines. Thus, they are closely tied to good configuration management.

The system requirements review (SRR) is used to review the adequacy and completeness of the draft functional configuration identification (system specification) for the system before establishing the functional baseline. The system design review (SDR) is used to review the adequacy and completeness of the draft allocated configuration identification (development specification) for each CI before establishing its allocated baseline. The preliminary design review (PDR) addresses the contractor's design concepts and preliminary design studies, prior to proceeding with the detail design, in order to assess his chances of meeting each CI's performance requirements. The critical design review (CDR)

addresses the contractor's detail design drawings (or programmable flow charts) prior to releasing them for manufacture/coding of qualification test (preproduction) articles; again, the object is to assess his chances of meeting each CI's performance requirements. At each of these design reviews, an additional increment of system/CI documentation has been generated. The purpose of each review is to detect and correct errors in this increment of documentation (and the contractor's design) before further CI development is undertaken based on the content of this increment.

The audits, to use Webster's definition, are a check of "the final statement of account" of the development program. The functional configuration audit (FCA) is used to check the results of each CI's qualification testing in order to verify that CI performance meets or exceeds specification requirements. The formal qualification review (FQR) is used to check the results of the system-level qualification testing in order to verify that all CIs meet or exceed their performance requirements as a part of the integrated system. The physical configuration audit (PCA) is used to verify the adequacy of the design documentation generated for each CI during the full-scale engineering development phase before establishing and product baseline and to verify the ability of the contractor's engineering (change) release system to control the release of changes to the product baseline documentation. The audits are used to verify the quality of the products of the full-scale engineering development phase prior to design approval and production authorization.

Configuration Control

Configuration control means the control of the baseline documentation. Contrary to popular belief, configuration control procedures, including the use of formal engineering change proposals (ECPs), must be implemented once the functional baseline (system specification) has been established. As the allocated and product baselines are established, formal ECPs must usually be approved before changes can be made to the baselined documentation. (Class II engineering changes to product baseline documentation are an exception to this approval requirement.) But in all cases, government concurrence with the proposed changes is required before theymay be implemented.

Configuration control requires that certain information be provided in the ECP to completely document all impacts of the change. Early in the development, the ECP content is relatively simple. It will describe specification wording changes, describe changes in the test program that result from the specification changes, and, in some cases, describe the general qualitative impact of the change on the logistics support and the operational capabilities of the system. During the production/deployment phase, a detailed description of changes in part design, of requirements for retrofit/rework of already delivered items and of impacts on

the logistics support system (spares, manuals, tools, etc.) must be included in order for the program office to assess the total impact of the change. The bottom line of configuration control during production and operation is to ensure the continued logistics supportability of the system once the change is approved and implemented.

Configuration control is not just something which must exist between the government and contractors. Once the production is complete, the contractor disappears from the picture. However, there is still a requirement for the government to control its own changes to the delivered items, whether they are the result of an approved modification program or an unauthorized maintenance action. Channels for official documentation and approval of all of these changes must be established if configuration control is to be maintained for the delivered items.

Configuration control implies the management of the flow of changes so that the program office will not be deluged with ECPs. It also implies preliminary communication between the program office and the contractor to try to reduce the costs of preparation of unwanted or incomplete formal proposals.

Configuration Status Accounting

Status accounting is probably the least understood part of configuration management. It is often perceived as a group of very expensive and voluminous reports used to track the implementation of approved changes. Large programs need them and pay for them; small programs can't afford them. Actually, however, status accounting is a management process; the reports are only one means of accomplishing this process.

Status accounting requires that all changes be carefully tracked from the time the idea is first recorded officially (in an advanced change/study notice, contractor or program office letter, preliminary ECP, formal ECP, or other document) until the time it is disapproved, or approved and officially incorporated into the contract. The intent is to expedite the processing of changes and to ensure that changes are not lost or delayed during processing.

Status accounting also requires the tracking of the implementation of approved changes after they are placed on contract. In the ECP, the contractor has identified all impacts to the program documentation and has provided a schedule for incorporating all changes into the production line, the operational units, and the logistics support system. Status accounting requires tracking of all these actions to make sure they are accomplished as proposed.

Most programs require an up-to-date record of the identification numbers and document numbers for each CI. They also require tracking of all approved ECPs for each CI and the production incorporation point (serial number effectivity) of the change. They need to monitor the development of new/revised manuals,

spares, and support equipment due to a change to the product baseline. For retrofit/rework changes, they would have to track the development and delivery of the kits of modification parts and the associated installation/checkout instructions. In most cases, they would also want to record the actual installation of these kits in operational units.

But whether all of this tracking is accomplished by reviewing formal reports, by weekly phone conversations with the contractor, by government plant representative checks, or by a combination of all of these, the tracking of the implementation *must* be accomplished. The amount and type of detailed information required for your program is your decision; the means of tracking will be determined by your program, but the tracking must be done.

Status accounting also requires the continuous tracking of the configuration of all units in the field. This is usually accomplished by the government agency. In some cases, this requires only part number control of the components installed in a configuration item or system; in others, there must be part number and serial number control of certain critical components. Differences in the configuration of like items must be documented and tracked; otherwise, a proliferation of configurations, e.g., local fixes to design deficiencies, can invalidate the program documentation and complicate the logistics support system.

During qualification testing, there is a need for the contractor to continuously track the configuration of each test item, even though the government has not established a product baseline to control the design. This tracking is necessary in order to verify the final configuration that passed the qualification testing, and to verify that all testing was accomplished on that configuration or a similar one. Otherwise, it will be difficult to validate the testing and to verify that the configuration item has met its performance requirements.

Who Needs Military Standards?

As a program manager who applies configuration management, you should understand it as a management philosophy and should understand why there are requirements for certain events, documentation, and milestones in your program. The actual implementation of configuration management will be determined by the program type, phase, expected production, manning, etc. The key is to accomplish the intent of the configuration management activities as you proceed with the program even if you are unable to accomplish the complete scope of each of the activities. Care must be taken, however, in deviating from the basic requirements spelled out in the military standards.

The military standards that describe configuration management program requirements for contractual application have been written and revised based on problems encountered and lessons learned in other DOD programs. The re-

quirements in the military standards may be tailored to delete unnecessary detail from the program, or they may be waived and entirely eliminated from the program. Indiscriminate tailoring or deletion, however, may lead to insufficient documentation and control of system design and may create communications problems in the understanding of contractual tasks.

The standards have established a common level of understanding of configuration management terms between the government and most DOD contractors. If I speak of a critical design review, the contractor understands the intent of the review and knows where to go (for example, MIL-STD-1521A) to find additional detail on what type of documentation will be needed and what activities will be accomplished. If I speak of a technical documentation review, he will need a detailed description of it in his statement of work and even then may not fully understand its purpose. And there can be problems if I require a "requirements specification" or use a "technical exhibit" to specify the performance requirements for a CI rather than, for example, a prime item development specification in accordance with MIL-STD-490. I may not include all of the necessary requirements (or there may be too many) needed to control the contractor's design process while permitting him enough design flexibility. And I may leave out some of the qualification tests needed to verify the adequacy of his design.

The military standards provide for a common understanding of contractual tasks between DOD and the contractors. The tasks associated with the standards are well defined and understood. If this is the first government contract for a contractor, the use of the military standards will give him a precise idea of the scope of effort required. The tailoring of these military standards for configuration management should be accomplished by a configuration manager who understands the needs of the program as well as the philosophy of configuration management. The same is also true for the selection of data items necessary to accomplish or document the tasks required by the military standards.

DOD-STD-480A (and MIL-STD-481A for some reprocurement contracts) establishes the requirements for submittal of ECPs, deviations, and waivers. It also defines the amount and type of information that should be included with the documents. MIL-STD-490 defines the criteria for selection of various types of program-peculiar specifications for use as program configuration identification. It also contains an appendix for each type of specification that defines the details of requirements that must be included in the specification. MIL-STD-1521A (USAF) is currently an Air Force-only standard, but it is also being used for guidance purposes by the other DOD agencies and is being modified for approval as a joint-services standard. It spells out the details of what must be accomplished at each of the various design reviews and configuration audits. These are the primary configuration management standards, and their tailored application to various programs must be accomplished very carefully. A tailoring guide has

been issued for MIL-STD-1521A, and one is in coordination for DOD-STD-480A.

In addition to these primary standards, there are many associated documents. DOD-STD-100C and specification DOD-D-1000B are available for guidance in obtaining engineering drawings and associated lists for the programs. Specification MIL-S-83490 establishes criteria for deciding the degree to which program specifications must meet MIL-STD-490 requirements. MIL-STD-499A includes requirements for the contractor's system engineering management program. MIL-STD-961 provides the content requirements for military (e.g., MIL-Q-9858A) specifications. And the Air Force has MIL-STD-483 (USAF) to supplement the requirements in the primary standards and to provide contractual requirements in areas not covered by the primary standards.

Problems for the Program Manager

Given this brief description of configuration management, you can see that it is a common-sense approach to a critical area of system program management. By incorporating adequate configuration management into your program, you ensure that you will be able to define and verify the configuration of the items and the logistics support elements you are buying; to control changes to these elements; to monitor the actual implementation of these changes; and to track the configuration of all units in the government inventory.

The problem is, where will you find the person with the qualifications to accomplish these configuration management tasks for you? For many years, configuration managers have been "configuration recorders," required merely to track and record the accomplishment of various program activities without participating in the initial decisions about the tasks. The configuration manager has acquired a reputation as a "paper pusher" and a "stick in the mud." It is true that part of his responsibilities do require administrative documentation of various program activities. And he does seem to be the program office conscience, or policeman, requiring complete review and official decisions before changes to the contract are authorized.

But the configuration manager all too frequently receives sparse recognition or praise for his efforts and management expertise. The resulting feelings of low esteem and low status compared to other functional managers have led to a perception of poor career progression and low promotability within the configuration management field. So the configuration manager will spend 1 to 2 years in configuration management learning about program management. When he has learned what he needs to know, he will transfer to some other functional management area where the promotion potential appears to be better.

If there is to be effective program management, there is a need to reverse this

trend. If we are to retain our best talent, we must establish an effective career progression for the configuration manager. The responsibility for program recommendations in the configuration management area should be delegated to the configuration managers; they should participate in the program planning sessions coequal with the engineers, test planners, logisticians, and other functional experts. The experienced configuration manager has the expertise in his functional area, the familiarity with most other functional management areas, and the insight into the program needs necessary to construct a suitable configuration management system. But, if he has transferred to another program office in some other functional management job, he won't be able to help you in your initial program decisions. And you can't expect a brand new configuration manager to have the necessary understanding.

There is a need to familiarize system program management personnel with the philosophy of configuration management. (For that matter, there is a need to familiarize all program management personnel with all the functional management philosophies.) This will help dispel their misconceptions about the effects of configuration management on, and the role of the configuration manager in, their day-to-day activities.

Also, since the tracking of the actual configuration of operational units requires the input of information from operational maintenance personnel, there is a need to address the reasons for and effects of configuration management and configuration control in the various maintenance training courses these people attend. We need to instill in them an appreciation of what increased configuration (proliferation) control can do to decrease or simplify their workload. Otherwise, we will continue to have control problems at the operational level.

If we are to have good configuration management on our programs in the future, we need to take action now to better employ the available resources of configuration management expertise. This is the only way that we can enhance the effectiveness of configuration management throughout the program life cycle. To continue as we have is to invite continued problems with the support of operational systems. ‖

Configuration Management in the 1990s and Beyond

30

Alan E. Lager

In the post-1990 period, computers and computer software, using yet-to-be-developed data base transfer techniques, will play a major role in every aspect of configuration management. Skilled professional configuration management personnel will be dealing with versatile, sophisticated equipment whose characteristics are a blend of hardware and software technology. With most design and modification being performed on interactive computer terminals, it will be difficult to tell where software ends and hardware begins. Like it or not, configuration managers will require software management skills—even to manage hardware.

They won't throw away all their pencils and paper, but advanced computer technology will absorb many present-day mechanical functions. Most configuration-related data will be software-generated by-products. Configuration managers, freed from most of the clerical aspects of configuration management, but with complete and precise information available to them, will play an important role in program management decision-making.

The challenge to today's configuration management practitioners is to look beyond the limits of today's techniques, else they risk becoming as anachronistic as the fabled bookkeeper with the green eyeshade and quill pen.

Our world is changing with ever increasing rapidity. What I think we can see, if we really look hard enough, is that some of the things on which we are focusing a great deal of attention today are becoming obsolete before our very eyes. While we concern ourselves with, among other things, engineering reports, specifications, types and forms of drawings, deferred data procurement, and even engineering change proposals, there is a distinct possibility that, for a large segment of industry, these and other present forms of data *may not exist* in 1990!

With the accelerating advance of technology, devices being produced today were unheard of just a few years ago. Today, I can put into my shirt pocket, or even hold on the tip of my finger, the computing capacity that only yesterday filled a building. Design and manufacturing methods are evolving with the use and assistance of the computer. Computer Aided Design and Manufacturing

Author's Note: The material for this paper was compiled at the Electronics Industries Association G-33 Committee's Workshop, held in October 1976 in San Antonio, Texas. It was my pleasure to be co-chairman of the panel entitled "The Current Evolution and Future Role of Configuration Management." I would like to gratefully acknowledge the contributions of each and every member of that esteemed panel, "experts" in every sense of the word.

Alan E. Lager is Manager of Configuration Management Technology, Grumman Aerospace Corporation, responsible for the planning of new configuration management projects and the implementation of state-of-the-art techniques, policies, and procedures. He has more than 20 years' experience in configuration management and related areas, and his 1968 paper, "Configuration Management—Its Role in the Aerospace Industry" (with Edward G. Hantz), was adapted as a chapter in the textbook, Fundamentals of Configuration Management, *by Czerwinski and Samaras. Mr. Lager holds a degree in mechanical engineering from Stevens Institute of Technology.*

(CADAM) is here, is maturing, and is rapidly becoming the way of life in many of our industries. Throughout all of this evolution, there is the ever present, increasingly powerful, and always changing technology of the computer. Computing hardware is getting smaller and cheaper, and each year we are doing more with the software than we ever thought possible.

Functional Models Supplementing Specifications

Let's gaze into the proverbial crystal ball to examine the potential technological evolutions in configuration management that are likely to occur in the next 15 to 20 years as a result of, and in conjunction with, the expanded capabilities expected in computers and software. I will attempt to describe how automated techniques will replace much of today's product documentation; how automation will assist the "change board" decision process; and how status accounting in real time will act as a downward driver of cost.

As I look into my crystal ball, I see functional computer program models supplementing and, in some cases, completely supplanting, paper versions of well-known configuration identification documentation such as: system, configuration item, and material and process specifications; test plans and procedures; and engineering drawings. These mathematical models, representing the real-time configuration of the product to be produced, are developed initially for conceptual phase studies and are continuously refined throughout subsequent phases. They are created using a general-purpose modeling system that can be used to refine and incrementally expand the model.

An interrelated hierarchy of these models is created and refined as the requirements for a system evolve and are allocated to lower levels. Product definition and description suitable for analysis, production, maintenance, modification, and reprocurement are created by computer-aided design data processing techniques. Most parts are produced using computer-aided manufacturing techniques. At any instant in the process the deliverable hardware and software, of any given configuration, are directly traceable to software stored in computer memory. All associated documentation is in the form of stored intelligence, which is accessible in readable form from computer terminals, CRTs and, occasionally, in paper or microfilm form from printer/plotters.

The models are used to develop test procedures, and to evaluate changes through simulation. They provide interaction through computer-to-computer crosstalk between and among the associate contractors, subcontractors, government furnished equipment suppliers, and government facilities.

The deliverable data at the end of the process, a completely definitized model containing the total product description, is in the form of a portable data file rather than in hard copy. It represents all the data needed for maintenance,

overhaul, logistics, and reprocurement, and yes, is small enough to fit in my shirt pocket.

Documentation as we know it today is minimal in the nineties. Some percentage of the equipment of the '70s and '80s is still a part of the functional inventory, and has to be supported with techniques that are a mixture of what we do today and partially implemented 1990s methods. For new products, methods of design and management are constantly evolving, paced only by the state-of-the-art in processors and peripheral devices.

Automated Configuration Control Board

Adjusting the fine tuning on my crystal ball, I am now picking up what seems to be a Configuration Control Board (CCB) meeting. At the head of the table is a projection screen linked to a computer terminal. When a question is asked, a technician keys it into the terminal and the answer is immediately projected onto the screen. Automation has entered the change control arena. It's not a frill; functional design changes can be so rapidly effected that, by necessity, change decisions have to be made quickly. Otherwise, control of the product configuration is lost, along with consequent cost and inventory control and profit. Present-day paperwork cycles would be intolerable.

Automation in the change control and change board decision-making process involves the exercising of the identification math models by computer programs that perform change analysis and evaluation. It also includes interfaces of data bases containing government and industry inventory, status, and work-in-process records. Each change is simulated by playing it through the model prior to the decision on its implementation to determine almost instantaneously its total impact, including the effect on performance, compatibility, interfaces, logistics, reliability, maintainability, and true life-cycle cost.

Failures or deficiencies in production or test, field failures, and customer-directed actions serve as preliminary initiators for impending changes. Field failure, reliability, and maintainability information is computer-collected, analyzed, and provided in synopsized form for change action when predetermined thresholds are approached.

Design activities respond by exercising math models to determine potential fixes. Engineers enter the detailed design into an uncontrolled computer field in which their trial-and-error changes will not affect permanently stored design data. Once the preliminary engineering solutions are reached, configuration management convenes the change board for the decision process, and the computer links and audio-visual aids allow real-time evaluation of the change.

Computer-stored design data, including preliminary design change considerations, will be simultaneously in view for all CCB members, including remotely

located members who are tied into the process by telephone/video/computer consoles.

Many alternate design solutions, including the choice of not making the change, are analyzed to determine their impact. Each is subjected to a cost-benefit trade-off analysis. Often the change that on the surface does not appear to be the best solution ends up being the choice because simulation shows it to have the best cost performance ranking over the life cycle.

Government approval and funding of changes can also be accomplished in real time through the network of computer interfaces. At the appropriate time, government program managers can be tied into the computer network and presented with the recommended design choice, a synopsis of the decision process, and an analysis of the impacts and risk factors involved. An approval and authorization to proceed may be immediately received via the computer link. Even the implementing contract modification may be accomplished as part of the joint government/contractor CCB action. The contract terms and conditions and contract specification, as well as all lower-tier specifications, are integral to the contract data base. They can be viewed, evaluated, negotiated, and modified by the CCB; however, human nature being the same in 1990 as in 1979, this may take longer than the technical decision.

Computer-Aided Change Implementation

After change approval and authorization, the computer again plays a part in the implementation of the change. As we saw, preliminary changes made for CCB evaluation were accomplished by entering needed data into an engineering controlled portion of the computer's "revision segment." Contractual design remains undisturbed until change approval is received, at which time revision segment data will be brought in to modify the contractual base file.

Performance and requirement parameter changes will be permanently inserted into the identification model or equipment specification file. A change in, say, output voltage made to the specification will automatically update the corresponding voltage in lower-tier requirements.

Instructions for direct operation of machines, rework/replacement orders, and updates to manufacturing plans, work schedules, and crew loading charts will be initiated.

Parts changes will cause a computer search for the corresponding parts listed in drawings, operation, maintenance, and parts breakdown handbooks, provisioning lists, and test procedures, all resident in related data bases. Parts changes in the computer will also trigger procurement, stocking, and eventual issuance of changed items.

Status Accounting Comes of Age

With only a slight adjustment to the focus control of my 1990 crystal ball console, I see that configuration status accounting has finally come of age. It brings the "real world" of product operation, maintenance, and logistic support in close accord with configuration identification and change control. Once again, computer technology makes a major contribution. A solid data base of configuration data delineates the real-time status of all items in the inventory. Through computer interrogation, it provides the change board, as well as the logisticians and maintenance planners, with the facts and analytical tools for accurate evaluations, cost projections, and decisions.

As a result of advances in the state-of-the-art of data base crosstalk, and other remarkable methods of data communication, a closer contractor-customer-using activity team relationship has developed. Gone are the misunderstandings caused by lack of information and interpretive forecasting tools. The conflicts among program participants are now minimized because the improved data banks accommodate dynamic inquiries and provide accurate and incisive responses that are timely and complete. Early problem identification and enhanced visibility of current status force relationships to become more honest and open. As a by-product, they tend to highlight any areas of data overlap and redundancy that can subsequently be eliminated by mutual consent.

The 1990 status accounting systems are designed to capture and transmit reliable information by remote means while still providing security control for sensitive data.

Input techniques are not rigidly standardized. They allow individual methodology and procedures for loading the data base, procedures that are extremely flexible and adaptable to a wide variety of contracting and product circumstances. What is standardized, to a large degree, are the common data elements being entered into the automated data banks and extracted by users at either local or remote locations. The new technology has fostered data element standardization at the same time as format flexibility. In the 1990s, the current version (automated of course) of MIL-STD-482 (or its equivalent) is actually used!

The key change status data and as-designed configuration data are created as by-products of the automated design and change control processes and are integrated with other management procedures. Status data, schedules, and lead times are automatically calculated at every stage of the change process, from request for change through CCB schedules, decisions, and customer approval; and then release of the updated design from uncontrolled to active status in the machine memory. Very little paper is generated except for action-oriented reports. Key milestones are automatically interjected into program management's

schedules and work authorizations. Decisions and implementing orders are conveyed to those in need of the information and, where required, automatic stop-effort notices are produced for areas whose in-process work is impacted by pending changes.

Status accounting as-built input methods cover the spectrum from handwritten punched cards to techniques such as optical scanners that monitor hardware configuration and provide on-line updates. Some of the newer software-oriented equipment contains within its memory a status accounting record of its own specific configuration. This configuration record is remotely accessed and modified as changes are implemented or as parts are replaced by maintenance actions. In these later circumstances, total traceability of configuration from the shop floor through the using activity's operational environment can be accomplished by interrogating the appropriate data base. The keys to this traceability are machine-assigned unique serial number identifications or signatures of all the parts and assemblies down to the replaceable part level. Specific signatures for software versions are also used to control the configuration of highly modularized software products.

Exercising Control

These new computer techniques enable hardware and software to be controlled throughout the development, production, operational, and maintenance phases in a continuous coordinated fashion. Change activity triggers a closed-loop system that follows a change until physically incorporated, whether in production or by retrofit by the user.

An automated search of current as-built data base records provides an immediate and completely accurate assessment of the configuration of delivered systems. The optimum serial number for both production and retrofit of a change can thus be determined at the CCB, as well as a quantitative analysis of the number, type, and installation point of the modification kits required.

Field modification data and kit parts lists are by-products of revision of the basic design data and will be generated by the computer models. Dependent upon the capability and equipment at the scheduled installing site, the instructions may be transmitted by hard copy, or directly to remote computer terminals.

As modification kits are delivered and then issued, as-built data are appropriately annotated to provide an automated suspense record. Modification kits include computer-coded forms, cards, chips, or mark-sense devices that are used to feed back incorporation data. The field installer checks the completion block and returns the data to update the file either by prepaid mailer or by accessing a remote reader tied to the computer network. Since the incorporation action is scheduled via an interfacing automated system, a "tickler" is produced if the required response is not received in time, and "past due" notices and delinquency

reports are sent to the installing activity and appropriate command echelons after a suitable grace period. This system functions with involvement by both contractor and government data banks. It is also used to control periodic maintenance and other "time compliance" requirements.

The volume of information that is compiled in this data base network is immense. Yet each user will be able to extract the information important to him in real time and in the format most usable to him. The new flexible reporting techniques allow us to ask the computer for any explicit set of related data elements, and a logical analysis of that data. We can, for example, ask for trend charts, statistical studies, financial summaries and the like, and we can get our answers almost instantly. The enormous memory capability and ultra-high-speed processing of the 1990s' computing hardware make these and other intelligence extremely inexpensive. The commitment to on-line systems and visual displays makes "zero-base" paper flow a practical reality.

How Do We Get There From Here?

Now that I've described what some may call a configuration manager's utopia, let's come back to the present and look at some practical realities of the year 1979. Is this future I'm projecting really possible?

Well, in some large, advanced contractor organizations there already exist systems that have begun to approximate components of the general system that we have postulated. We've already spoken about the current advances of computer-aided design and manufacturing. Some automation of the change control process has begun at several companies, and several have developed significantly automated status accounting and verification systems. The growth of these concepts and their acceptance by corporate management over the next dozen to twenty years is, of course, greatly dependent on the rate of advancement and the subsequent application of new computer technology.

Economics, as always, will be the primary driving force behind a gradual evolution effected by industry as a whole. Impetus will be provided in individual large-scale government programs. As present-day engineering and management personnel are gradually replaced by future college graduates whose whole educational process has been influenced by the computer, the acceptance of widespread computer use in all areas of business will be accelerated. The new breed of manager will have been "weaned" on the computer and will not have the mistrust that still prevails among many in top-level management today.

I think that government action will be necessary, and will be forthcoming, to influence the development of systems in the configuration management area. Of course, the large-scale aerospace programs will lead the way, and perhaps a series of government-funded pilot programs and studies will be established by the serv-

ices. These will lead to more standardization among the services so that common techniques can be used by all. The concept of "lead service" procurement in which one service acts as the purchaser for a given commodity, is a precursor to standardization of systems. Although slow in its initial stages, closer coordination between services will become an economic necessity.

Will the new technology leave the little guy out in the rain while the giant corporations and the government agencies play their computer games? I don't think so. Current cost trends in processors and peripheral equipment, and in leased-line microwave communication networks, strongly imply the accessibility of the new techniques to businesses of all sizes.

As far as configuration managers go, it doesn't take a genius to realize that we had better get software-oriented and learn how to manage software—and fast! While the new "support" software systems are being created for automated design and manufacturing of hardware, and "software factory" techniques are applied for the development of software, we must take an active part. Unless our systems of identification and control of configuration are built into, and integral with, the support software, we will not be able to keep pace.

If we continue to conceive of doing the job as we are today in spite of technological advances, we will force excessive and unnecessary costs to be incurred and excessive work to be layered on top of the basic on-machine design and production efforts. The new computer-minded managers will be quick to see the inefficiency of the "paper pushers."

A New Frontier

If we look into the future, we see a new frontier in configuration and data management. Our challenge is to define it and apply our innovation and imagination to the solution of its problems. We must evolve with the times by using the technology of tomorrow to manage the products of tomorrow. If we don't, we are in danger of becoming obsolete.

The message is clear. We must promote today the discipline, tools, and caliber of personnel required to manage tomorrow's sophisticated world of configuration and data management.

Ensuring an Adequate Technical Data Baseline

38

John R. Hart

A baseline, in general, may be considered a foundation, or basis, upon which something derives its support, or against which certain activity is measured. The rainbow could be considered a baseline for identification and correlation of the multitude of available colors; the Constitution, of course, serves as the baseline for the enactment and adjudication of our laws; and mortality tables form the basis for determination of insurance costs.

There are any number of baselines associated with the development of an item or system, including technical data baselines. The most common technical data baseline in the aerospace field involves the configuration definition of an operational item or system. This definition finds its outward expression in a set of operational drawings and specifications. However, experience has shown that a total representation of the technical data baseline requires the availability and understanding of many of the elements that are shown elsewhere, e.g., tooling drawings, parts substitution criteria, etc. In addressing the matter of a technical data baseline, this paper will consider the system configuration definition in general, recognizing that the same or similar observations would apply to almost any technical baseline.

Baseline Integrity

Establishing technical data baseline integrity generally involves identifying the technical requirements and then verifying that the resulting data base actually satisfies those requirements. In the government/contractor interface activity, the accepted technical data baseline is identified, maintained, and controlled via the contract, which essentially covers the delivery of hardware (and software); the content and delivery of data; and/or the accomplishment of certain services. The principal contract elements, the statement of work, the contract data requirements list (CDRL),* the systems specification, the general provisions, the identification of applicable documents and the application of the Defense Acquisition Regulation (DAR—formerly Armed Services Procurement Regulation, or ASPR) clauses all may invoke requirements affecting the technical data baseline through acquisition management systems (both the hardware and the non-hardware types), data item descriptions, or other government-generated constraints.

Confidence in the integrity of a technical data baseline can only be as strong

*A brief definition of this and other terms used throughout this paper can be found in the "Definition of Terms" section immediately following the article.

John R. Hart is the Assistant to the Information Systems Manager at Boeing Aerospace Company. He previously held assignments in engineering design, systems engineering, technical requirements, and information systems management in Bomarc, Saturn ASMS, and Minuteman programs. Mr. Hart holds a B.S.E.E. degree from the University of Utah.

as the confidence in the methods and procedures used to generate the information from which the baseline was established. Only when a high degree of confidence in the baseline is achieved can meaningful verification be realized.

Primary and Secondary Baselines

Consider the data involved in establishing a configuration definition baseline. The primary (ultimate) baseline is derived from drawings, specifications, and technical manuals/handbooks/procedures. The secondary (initiatory) baseline is generated through reports, analyses, records, forms, plans, studies, lists, documents, forecasts, notifications, evaluations, diagrams, etc. The primary baseline, of course, establishes the standard that becomes the basis for verification. The secondary baseline confirms that the primary baseline is valid. The process of establishing and utilizing these baselines is an ongoing one that may be considered as verifying (1) the integrity of the contractor's participation in the endeavor, (2) the satisfactory accomplishment of the efforts that generate the two baselines,and (3) the integrity of the primary, or ultimate, baseline. The major portions of a contract are designed to accomplish this verification.

Impact of Contract Makeup on Baseline

In making this whole system play together, the government (and industry) generally assigns someone the responsibility for generating (responding to) the statement of work and applicable documents definition; someone else the system specification; someone else the CDRL; and someone else the general provision/DAR definitions (or variations of these assignments). It should not be too surprising, therefore, that (possibly as a result of this fragmentation) the many factors in a contract that affect a technical baseline are found in all contract elements.

The verification of both the secondary and primary baselines is complicated and visibility made difficult under the present systems because of the:

- Large volume of tasking documents that impact baselines;
- Large volume of data-generating instructions, many of which are duplicative or inconsistent with each other and hence confusing to a contractor dealing with more than one government office;
- Contract disparity;
- Contract internal inconsistency;
- Inordinate requirements imposed but semi-visible (DAR, multi-tier reference); and the
- Attempt to control data product separate from task.

TASKING DOCUMENT VOLUME

There are available for use at least 121 different types of tasking documents (and tens of thousands of individual tasking documents), only a small percentage of which are controlled to any extent (e.g., specifications, standards, OPNAVs, ARs, AFRs, IIPs, KAWs, DSAMs, etc.). Although, depending upon the DOD element involved, different sets of these tasking documents are applied in a given contract, it can be expected that each and every one has some impact on a technical data baseline. These requirements are not always consistent and many conflict with each other and with higher-level requirements.

LARGE VOLUME OF DATA-GENERATING INSTRUCTIONS

A review of the latest issue of the acquisition management systems and data requirements control list (AMSDL) shows 34 different acquisition management system documents dealing in one way or another with configuration, and it is evident that there are many, many more that have an impact on configuration but that do not show up as such in the AMSDL. Ninety-three data item descriptions (DIDs) are specifically tied to the acquisition management systems listed under configuration. The AMSDL also identifies some 773 data item descriptions that derive from the 34 acquisition management systems.

Figures 1 and 2 (excerpted from the lists of DIDs tied to the source document MIL-STD-480 on engineering change proposals and requests for deviations and waivers—current AMSDL) illustrate the overabundance of direction on the same

FIGURE 1
MIL-STD-480 Configuration Control—Engineering Changes

DID	INFORMATION
E-1102	Engineering Change Proposals
E-2037	Engineering Change Proposals
E-4527A	Engineering Change Proposal
E-5035B	Engineering Changes
E-6204	Exhibits, Engineering Change Proposals
E-21351	Summary, Engineering Change Proposal
E-23101A	Proposals, Engineering Change
E-23435	Proposals, Engineering Change
E-25603	Trainer-Engineering Change Proposal Summary
E-3128*	Engineering Change Proposals
E-2177	Software Change Proposal
E-2038**	Engineering Change Proposals
E-5034B	Engineering Changes (Short Form)
E-5383A**	Engineering Changes Commercial Format

*MIL-STD-490 Source Document
**MIL-STD-481A Source Document

subjects that is available to be imposed on a contractor, much of which may be essentially the same in intent but vary substantially in makeup.

Not all of these constraints, of course, are invoked on any given contract; nonetheless, the effort involved in establishing the integrity of the requirements that generate either the secondary baseline or the primary baseline can be readily seen as being extremely large and complex. Further, any change from one constraint to a "similar" one has a negative effect on the confidence level built through use of the former constraint.

CONTRACT DISPARITY

It is interesting to note that different DOD elements interpret regulations differently, and these differences appear in ways that affect, at least, the secondary baseline. This means that, ironically enough, experience in reading contract packages from one DOD element may actually make it more difficult to understand and respond properly to contract packages from another agency. Consider the circumstance where one system program office (SPO) is rigorous about using standard DIDs. After some experience in this environment, a contractor receives from an SPO a contract package that is standard only in its use of DID nomenclature. Over the period of contract performance it is inevitable that confusion will develop concerning what the data requirements actually are, and this impacts the confidence level normally afforded by the secondary baseline.

FIGURE 2
MIL-STD-480 Configuration Control—Deviations and Waivers

DID	INFORMATION
E-2037	ECPs and Requests for Deviations and Waivers
E-3129	Request for Deviation/Waiver
E-5035B	Engineering Changes, Deviations and Waivers
E-20134	Change, Deviation and Waiver Form
E-23102A	Deviations, Request for
E-23103A	Waivers, Request for
E-5034B*	Engineering Changes, Deviations and Waivers
E-5035B*	Engineering Changes, Deviations and Waivers
E-20134**	Change, Deviation and Waiver Form
E-2038***	Engineering Changes, Deviations, Waivers
E-5383A***	Engineering Changes, Deviations, Waivers
E-2038***	ECPs and Requests for Deviations and Waivers

*MIL-STD-483 Source Document
**MIL-STD-490 Source Document
***MIL-STD-481A Source Document

CONTRACT INTERNAL INCONSISTENCY

Figures 3 and 4 show the results of the analysis of the makeup and internal correlation of a recent DOD contract and a current request for proposal (RFP). As can be seen, in terms of the establishment of the requirements that result in the generation of both the secondary and the primary baselines, these contract packages are remarkably out-of-sync internally. Again, this impacts the confidence level of the secondary baseline.

INORDINATE REQUIREMENTS IMPOSED AND SEMI-VISIBLE

Many requirements that affect both the secondary and primary baselines are not self-evident. That is, one might reasonably expect that all of the AMSs that the contract applies would be listed in the applicable documents section, and that all of the requirements for data would be included in the CDRL. As illustrated in Figure 3, this is not so. Following the referenced requirements tree for data to only the third tier results in a quantity of "requirements" that is an order of magnitude larger than those called out on the CDRL, ostensibly the sole source of data requirements on the contract. A similar situation exists when tracing AMS/specification/standard requirements to the third tier.

Figure 5 illustrates an additional source of "non-visible requirements"—the DAR (ASPR). The number of clauses having data impact is roughly equivalent to

FIGURE 3

Contract Package—Current RFP

AMSs		DIDs	
Total AMSs Specified	152	Total DIDs Specified	106
Hardware Impact Only	5	On DD 1423	57
Cost Driver Areas (Shea Report)	147	Not in DD 1423	49
Identified in DODISS	79	Third Tier References	2,937
Not in DODISS	73	Number of Data Requirements	
Invoked via System Spec. (S.O.W.)	85	Not Identified in	
Invoked Via Data Requirements	67	S.O.W./System Spec.	49%

	S.O.W. APPL. DOCS.	SYSTEM SPEC. APPL. DOCS.
Total	88	34
Common	30	30
Missing	58	4

(Also includes conflicting requirements, e.g., drawing prep. MIL-D-1000A, DOD-D-1000B.)

the number of sequences on the CDRL, and although probably not of major importance to the primary baseline, is certainly a factor in establishing the integrity of the secondary baseline.

ATTEMPT TO CONTROL DATA PRODUCT SEPARATE FROM TASK

For many years, DOD and industry have been chugging away on a most well-intentioned, apparently worthwhile task—the elimination from DOD contracts of unnecessary data. The CDRL was developed to help accomplish this goal; data pricing requirements were established; tracking systems (such as AMIS—acquisition management information system which, in itself, generates additional data) were set up or modified to keep track of data performance; SPOs began to require identification and justification of all CDRL data, etc. But all of these things essentially have been done on the premise that the data product can be identified, priced, cost-benefit-analyzed, and managed separate from the generating source document. It is my belief that as a result of all this activity, a substantial amount of DOD funds are wasted every year in meaningless data pricing. We hear complaints about the high cost of data and yet find that the costs specifically identified to "data" are small. Additionally, the mismatches cited earlier between data requirements and AMSs continue to occur. Furthermore, we attempt to control the generation of data, but are essentially oblivious to what is being done toward controlling the corresponding creative task effort. As a result, we find ourselves frustrated trying to manage and control data products while someone else is managing and controlling those tasks that generate the data.

FIGURE 4
Contract Package—Current Contract

ANALYSIS:

AMSDs Cited in S.O.W.		45	
AMSDs Cited in Annex 1-A		24	
DIDs Related to Source Doc	41		
Unique DIDs (No AMSD)	17		
DIDs With No AMSD	6		
Total CDRL Sequences		64	
S.O.W. AMSDs Cited in Annex 1-A		14	
CDRL AMSDs Cited in S.O.W.		6	4 Common
CDRL AMSDs Cited in Annex 1-A		6	
AMSDs Tailored in Annex 1-A		5	

New DOD Program

Now, for some time, DOD (Office of the Under Secretary of Defense) has been endeavoring (with surprising resistance from some elements in both DOD and industry) to do an interesting thing to contracts—tie together the data product with its generating source document, contractually, and restructure the contract makeup to clarify this relationship and the total AMS/data requirements imposed. Interesting things can happen when this modification of the current system is utilized:

- The office of prime responsibility will have visibility and will control all data requirements associated with his task document, thereby controlling also the proliferation of DIDs.
- *All* requirements affecting the technical baseline (DAR, public law, AMSs, DIDs, etc.) will be identified centrally so there are no hidden requirements.
- The contract package can be expected to have improved internal consistency. When all of the contract constraints (AMS/DID/DAR, etc.) are summarized centrally, the establishment of each baseline can be done uniformly and with confidence.
- The true data cost—both the preparation (task) and the publishing efforts (data)—can now be addressed (if desired).
- The impact of the imposition of DAR and public law constraints can be dealt with (if desired).
- The AMSDL can realistically be expected to shrink through elimination of duplicate DIDs and thus can become a more effective management tool.
- The way can be paved to allow the SPOs to manage those elements best managed at their level, without the necessity for having each decision approved at higher command levels.

FIGURE 5
Contract Package—Current RFP

Number ASPR Clauses Cited	**145**
(Public Law Requirements)	**(13)**
Number Having No Data Impact	**21**
Number Having Data Impact	**124**
Potential/Probable Submittals	**58**
Generate Data Records	**35**
Specific Data Submittals	**25**
Applies Other Docs.	**6**

As a result of this approach, we find that, first, the requirements for each technical baseline can be more clearly identified. The uncertainties as to how many additional requirements are contained in cited DAR clauses and lower-tier references can effectively be eliminated. All requirements impacting the baseline are identified together in the contract, rather than being scattered throughout the various contract elements.

Second, our confidence in the secondary baseline can be more readily achieved. Internal consistency in contracts clarifies the effort to be accomplished; provides for better cost-benefit need analyses of contracted efforts, and facilitates the transfer of confidence in contractor management systems from one prime DOD element to another.

Third, the integrity of the primary baseline can be more significantly assured. The task source document office of prime responsibility is in the very best position to ascertain which data are required to satisfy the needs of that source document. Funded effort can therefore be expended to directly support the generation of an optimum baseline rather than to include many of the "nice-to-have" items that now get funded.

The result of all of the above can be expected to be improved confidence in the secondary baseline and an enhanced assurance of the integrity of the primary baseline. Although much progress has been made, the full benefit of this effort cannot be expected to be realized without:

- The literal acceptance by DOD elements and industry of the policies and procedures already identified in such DOD media as DODD 4120.21, DODI 5000.32, and DODD 5000.19.
- The expanded use of tailoring and contractor management systems.
- The issuance of implementing media to carry out the policies already established.

These constitute the major challenges associated with the overall objective of assuring a technical data baseline through administration of management information systems.

Definitions

Acquisition Management Information System (AMIS)
An AFSC-developed system designed to implement source data automation. Details of contracts are computerized and made available to users and status/summary information is made available upon suitable inquiry.

Acquisition Management System (AMS)
A document contract requirement that directs or constrains the manner in which the contractor achieves the end product of the contract. It generally outlines a detailed procedure that describes an orderly way of assisting managers in defining or stating policy objectives and requirements; assigning responsibility; achieving

efficient and effective utilization of resources; periodically measuring performance; comparing that performance against stated objectives and requirements; and taking appropriate action.

DOD Acquisition Management Systems and Data Requirements Control List (AMSDL)
A master list of approved acquisition management systems, data source documents, and data requirements to be used when applying such requirements to a given contract.

Armed Services Procurement Regulation (ASPR)
A regulation issued by the Assistant Secretary of Defense that establishes uniform Department of Defense policies and procedures relating to the procurement of supplies and services under the authority of Chapter 137, Title 10 of the United States Code, or under other statutory authority.

Contract Data Requirements List (CDRL)
A contract form, DD Form 1423, listing all technical data items selected from an approved data list required to be delivered under the contract.

Defense Acquisition Regulation (DAR)
Serves the same function as, and is in the process of replacing, the ASPR.

Data Item Description (DID) (DD Form 1664)
A form that specifies the data that must be furnished. The forms specifically define, using the descriptive method, the content, preparation instructions, format, and intended use of each data product.

Office of Primary Responsibility (OPR)
The DOD Component having primary responsibility for the acquisition management system/source document and/or DID.

System Program Office (SPO)
That DOD element charged with managing (administering) a specified DOD program or procurement. ‖

47 Aviation Configuration Accounting System

Leon W. Grzech

It is axiomatic that effective operation and management of aviation materials depends on current, accurate information for decision-making. Concomitantly, the foundation for safety of operations and effective management of aviation equipment (configuration items) rests upon diligent application of disciplines in configuration management; namely, applying technical and administrative direction and surveillance to (1) identify and document the functional and physical characteristics of a configuration item, (2) control changes to those characteristics, and (3) record and report change processing and implementation status.

Present command policies and procedures implementing these policies are adequate for identification and control. While policies are adequate for CSA, or configuration status accounting (recording/reporting), available means for implementing CSA policies are, and have been, seriously deficient. The problem is that the CSA data presently generated are not only stratified throughout the aviation community, but the existing management information system lacks the means, short of labor-intensive processes, to identify, collect, sort, collate, and communicate CSA data in a suitable format for timely acquisition, operation, and logistic support management decision-making. There is an urgent need to make available CSA data, "keystone" to most Navy management information systems.

The project endorsed by the Chief of Naval Operations to correct CSA deficiencies has been formally designated "AVCAS," or Aviation Configuration Accounting System, which is a distributed data base network system. Conceptually, AVCAS will provide the medium to better organize the data collection, entry, processing, and application of Naval CSA information into a cognate communicable state.

Unlike most initiatives toward mechanization of management data to serve the semi-autonomous functions of managing procurement of hardware, AVCAS will capitalize on existing and programmed procurement of computer resources and associated data communication systems. When CSA is structurally embedded within the data base candidates for AVCAS means and methods, the individual functional effectiveness of those candidates is further optimized.

To achieve its objectives, AVCAS adds capability to those systems that inherently lend themselves to the incorporation of AVCAS means and methods.

Author's Note: I want to express my appreciation to the Naval Weapons Engineering Support Activity, Washington Navy Yard, for the engineering counsel and aid rendered to me during the preparation of this paper.

Leon W. Grzech is Configuration Status Accounting (CSA) Planning Officer at Naval Air Systems Command, Washington, D.C. His duties include CSA policy formulation and application, CSA systems development, and automated data processing applications. He is chief project architect for design/development of the Configuration Status Accounting System.

Uniting resources proposed in the AVCAS design will simplify the execution of the data-decision-action feedback loops involved in the complex matrix arrangement of Naval aviation management. This will require a high-density diversified data processing and transcommunication network. Fortunately, adequate resources are either operational or planned for implementation in a time frame compatible with AVCAS development. The requirement for AVCAS has existed since early 1972. Recent technological progress now permits us to pursue development of a distributed ADP system.

AVCAS—Early Development Constrained

Naval aviation management information systems (MISs) evolved as a loosely associated group of semi-autonomous functional systems, each responsible for supporting specific, yet distinct, administrative, management, operational, or logistics tasks. Management information systems were, for the most part, unique to the individual activity and were non-standardized except for the operating forces. The *modus operandi* peculiar to each was generally unfamiliar among "outside" activities. Processes and procedures were often developed for in-house requirements with little regard for the interaction of these processes and procedures with external interests and functions. CSA evolved in a similar fashion.

History has demonstrated that mankind's welfare progresses proportionally with his ability to communicate effectively. Communications via mechanized means, i.e., telegraph, telephone, and radio frequency transmission modes, have contributed greatly to our knowledge by providing more timely and accurate information. However, each of these media is subject to delays when management inquiries require the respondent to employ labor-intensive measures to search, identify, process, and transmit the source data contained in the respondent's files. Manual processing of large masses of data is giving way to the more economical forms available with each advancing step in both computer hardware, software, and communications technology.

Early Navy automated data processing (ADP) applications relied on tape-oriented computers in order to accommodate large masses of data. Data files were primarily archival. Moreover, data usage was virtually limited to statistical batch processing. Direct user inquiry of the file, i.e., via terminal equipment, was not economically feasible, and the file suffered benign neglect. Real-time random-access computers, a more economically prudent alternative, are being tasked to handle ever-increasing masses of data drawn from various sources. If judiciously integrated into networks, computers provide us an opportunity to make another communication breakthrough with resultant improvement in management capability.

The introduction of new computer technology has not been without its prob-

lems. Consequent to the introduction of competitive and more powerful hardware, particularly the minicomputer, a wave of computer applications evolved in order to meet localized individual requirements. The *force majeure* during this era was management's requirements for data timeliness and accuracy, which still persist. Computers, with their accompanying software routines, proliferated. Military considerations for standardization were in jeopardy. A move toward standardization of both hardware and software was pursued.

In some quarters, the cause for standardization suggested a need for centralization of the data base; indeed, the corporate memory. Proponents of data base centralization would suggest that redundancy of files would virtually be eliminated as a consequence of centralization. Such claims are without merit unless two or more separate data bases result in the production of the same reports. The fact that one or more data bases may rely on the use of similar, individual data elements and report forms (not actual content) is a matter of consequence to the similarity of management function(s) supported by each data base.

Advocates of decentralization, on the other hand, point to the very nature of the matrix organization, recognizing that: The function of each unit within the organization is semi-autonomous; that data generated in a logistically oriented environment involving custodial/sub-custodial resource accounts should be immediately available to the custodian; that information necessary to the upper echelons of command management is usually more succinct (statistical/graphical) in form; that unless the file can be maintained locally through direct access at the lowest level possible, it will suffer benign neglect; and lastly, that in the event of failure of either the computer, its facility, or its line power, a centralized data base would deny service to the individual members in the community for the entire computer outage period. This would be particularly crucial to independent military units in time of war. Conversely, outages in a distributed data base affect only that local system segment. Even in the temporary absence of the network communications, the individual segment is still functional. The degree of file degradation is proportional to the time of the communication outage period and amount of data to be communicated during this period. However, if the individual sites affected have machine-readable format capability, only timeliness of data among the network segments impacted is affected.

Application of resident/older ADP systems in the military could not satisfy requirements. This was largely due to untimely response of tape-oriented computer systems. A 40- to 60-day response, typical of these systems, was and is unsatisfactory for most user requirements. Moreover, this type of system appeared to make the data inaccurate, not because the user's transaction report was in error, but mostly because of time lag between subsequent changes to the original transaction report and the period necessary to communicate and process the data.

The data records are otherwise overtaken by more current events.

Improvements in computer technology now make it possible to resolve the present inequities in the information processing and reporting system used in support of the billions of dollars of weapons and support systems in the military inventory. Moreover, orchestration of the acquisition, operational, and logistics support configuration management data demands a unified data system employing standardized data elements and report forms.

The AVCAS project envisions the use of localized mass-storage, random-access computers to service independent user requirements where data generation and frequency of use dictates, while simultaneously allowing other unique data bases, whose improvement hinges on direct data access, to automatically draw down on information necessary for its operation.

Data Collection Methodology

An objective of AVCAS is to implement a method of identifying and describing configuration items in terms of end-use and application requirements.

AVCAS will utilize a hierarchical indenture scheme formatted to identify and describe configuration items at various levels of sub-system support. This indenture scheme will be designed to identify and describe configuration items in a top-down context from the end-item to the lowest level necessary for configuration status accounting. The recording and retrieval of data toward achieving configuration status accounting mandates, minimally, marking of a configuration item with the following: part number, manufacturer's code, and serial number. This information, in addition to related upper and lower assembly and other essential data, shall be contained in the AVCAS data base.

Approved Navy configuration changes to the product baseline (the initial approved or conditionally approved product configuration identification) may be incorporated at the manufacturing plant, the Naval Air Rework Facility (depots), the intermediate maintenance activity, or the organizational level of maintenance. The intermediate maintenance activity and organizational level of maintenance operate under the instructions given in OPNAVINST 4790.2, Naval Aviation Maintenance Program, and are subject to the inherent maintenance, material management data collection system requirements. The Air Force and Army have comparable reporting systems. The Naval Air Rework Facility and the commercial contractor each operate under various data processing systems. The AVCAS design study team investigated each Naval Air Rework Facility data system available. Recent initiatives by the Chief of Naval Operations (OP-51) indicate a move toward establishing fleet data interface with Naval Air Rework Facility information-generating systems. Moreover, NAVAIR initiatives have resulted in obtaining configuration status accounting data from the contractor

data requirements list in suitable computerized tape format.

The data base employed by each custodian/sub-custodian of configuration item(s) involving its operation and/or support will become standardized and, once automated, will lend themselves to a distributed data base network.

Generic File Structure

The AVCAS is conceived as a configuration status accounting system that will be equally applicable to aircraft engines, ground support equipment, trainers, and other selected items of material, e.g., missiles and pods, as deemed appropriate by proper authority.

The AVCAS system design will apply to all NAVAIR aviation material that is configuration status accounting sensitive. The AVCAS data base will accommodate all end item type-model-series systems identifiable by assigned or pseudo type equipment code.

The AVCAS data product array will produce outputs in each generic category, and in each configuration item group within generic categories.

Design Guidelines

Design guidelines emphasize the need for a cogent and systematic approach to improving the coordination of data flow from the initial engineering change proposal evaluation and approval process through the technical directive incorporation and reporting procedures. Many feedback loops must be closed as configuration change data travel from the acquisition command responsible for expedient end-item configuration change, through the data processing media, to the change-status reporting instruments. The AVCAS system design addresses the feedback loop completion requirements that will enable timely and complete responses to user inquiries.

Fundamentally, configuration change data must be created, collected, processed, and made available to the cognizant user and supporting activities. The data and file(s) must be oriented to the particular organizational function or action to be exercised as a result of the information content of the data. Ultimately, all appropriate decisions, functions, and actions must be coordinated to provide harmonious interaction and interface among user activities. Generally, the engineering change proposals become one of several types of technical documents, with acquisition, operational, and logistics support impact factors adequately evaluated and identified. Once the change is approved and authorized for procurement, kits are fabricated and made available to the activity responsible for kit incorporation, coincident with the availability of the end-item designated to undergo a configuration change. Lastly, the configuration change must be reported promptly, completely, accurately, and to the proper data col-

lection point(s), including deconfiguration actions for whatever reasons.

The AVCAS design guidelines provide for the creation of standardized data base and data base management system directed toward servicing even specialized user requirements. Furthermore, by utilizing data telecommunication networks such as the Autodin II, AVCAS will facilitate improved interactivity data communications. The real-time (on-line) distributed data base network, coupling interactive computer processors and remote job entry terminals, is central to the effective application of the AVCAS.

Functional Systems Sensitivity

Operational configuration status accounting data is primarily related to the performance and functional compatibility aspects of weapon system constituent sub-elements. The possibility of system performance degradation resulting from installation of incompatible components is a growing concern in the operating forces. The AVCAS design guidelines include providing to the operational commands the sub-system compatibility data applicable to specific end-item configuration tolerances.

Effective support of configuration items demands continuous and consistent development of configuration change data. Maintenance performance and the resulting weapon system readiness attainment are completely dependent upon viable configuration knowledge.

Logistics support is particularly sensitive to configuration change. Some areas are maintenance technique, training requirements, technical documentation update requirements, spares and spare-parts modification, support equipment to support and/or incorporate airborne system changes, and support equipment configuration changes dictated by end-item configuration changes. The AVCAS data products will provide for a coordinated logistics support file access capability that will service the configuration status accounting information needs of logistics element managers.

AVCAS is being developed to provide to the widest spectrum of Naval aviation user activities a viable means of identifying and describing, as necessary, each configuration item throughout the life cycle of the end-item. The following activities are considered users/potential users of AVCAS data: Chief of Naval Operations; Chief of Naval Material; systems commands; project/program management; assistant program manager for logistic support; Naval Aviation Logistics Center; industrial, manufacturing, and rework activities (organic/commercial); interservice activities (USMC/Air Force/Army); Aviation Supply Office and stock points, including Ships Parts Control Center (SPCC) for ship/aviation interface support; single wholesale and inventory manager; fleet activities to include type commanders, wing commanders, organization, intermediate, and

Supply Support Center; aviation/non-aviation ships; field activities to include Naval Air Technical Services, Naval Weapons Center, Naval Missile Center, and Fleet Analysis Center; foreign military sales cases (by direction).

Opportunities

The project is not complete when AVCAS becomes operational. A mutual contractor/government quest to simplify government contract data requirements levied within contracts may be realized once AVCAS is operational. Certain reports present companion traits and are candidates for revision toward the consolidation and reduction of data element(s) reporting. Other initiatives are possible once AVCAS and its associated data-base network become familiar to the users and exposed to the proper algorithm. They are (not inclusive):

- Parameters such as out-of-service-time and readiness condition status variables may be more precisely evaluated in terms of configuration-change requirement impacts on system availability and compatibility through AVCAS.
- Precise compilation of indices of readiness degradation cause-and-effect determinants are possible. Measurement of configuration change imposed resource consumption is often desired as one of several variables in cost effectiveness and value assessment trade-off analyses.
- Current baseline data, if combined with design change notice (supply data that identifies parts removed and replaced on a configuration item subject to change), will serve to improve the quality and timeliness of the aviation allowance lists used to provision and support the fleet air community both ashore and afloat. The processes of support material procurement and provisioning are among the most complex in the Naval aviation community. To be effective, these processes must receive current, complete, and accurate configuration status accounting data.
- Demand for a specific configured spare, compatible with installation requirements, can best be met through AVCAS. Cost-effective modification of spares and spare parts, as well as procurement of modified spares and spare parts, demands viable end-item configuration status accounting. The AVCAS data will be generated that identifies and describes the configuration item to be supported in data elements familiar to the provisioning and support activities. Included in these data will be evolutionary configuration status by individual population of serialized configuration items.
- AVCAS will greatly minimize the need for the frequent configuration item physical audits that are presently necessary to accurately determine current configuration status prior to workload planning for material requirements and resource allocations evolutions.

- Manually maintained logs, locally prepared worksheets, and physical disassembly represent the present modes of configuration status accounting and verification techniques in most Naval aviation operation and support facilities. AVCAS will create a means to automatically log, identify, and describe to the workload planner, auditor, and the inspector the current configuration status of a top assembly item or any of its sub-assemblies, installed or uninstalled. ‖

55 Observations on Configuration Management

Lieutenant Colonel William G. Fohrman, USAF

Although the definition of configuration management in MIL-STD-480 and AFR 65-3 relates to a management process, these directives, in particular AFR 65-3, address the task elements of configuration management (identification, control, status accounting, and audit) with little emphasis on management. In the eyes of many, and too often in practice, configuration "management" equates to merely performing these task elements. This perception reduces configuration management to a clerical function of participating in and recording events.

Unfortunately, there are many in the discipline who are quite content to be configuration recorders rather than configuration managers. If configuration recording is the true function, I submit that many configuration managers are overpaid. If a configuration manager limits himself to performing within the task confines of AFR 65-3, he is either not getting the job done, or he is leaving it to someone else.

I view configuration management as involving three elements: administrative, clerical, and technical management. As the clerical and administrative functions are well understood, I will limit my comments to the technical management element.

If one is to manage, one must have a resource to apply to effect a solution. That resource may be expertise, manpower, or authority. Ideally, it is all of these. An effective manager tends to influence the outcome of an event in a positive and productive way. If one accepts this work-a-day definition, there are few configuration managers who truly manage. The reasons are that the management aspect is not adequately defined in our basic directives; configuration management activities are not normally visible to top management; and the preponderance of configuration managers are not technically oriented.

Configuration management grew from and is still essentially a sub-discipline of engineering. While engineering is responsible for item performance, configuration management is responsible for documenting that performance. The configuration manager's main interface throughout the development of a program is primarily with engineering or the products of engineering. We have found that, generally, the technically oriented configuration manager does a better job than a non-technical person and, specifically, that the configuration managers who are engineers tend to do the best job. The reasons seem to be that there is no credibility gap when dealing engineer to engineer, and that the depth of understanding of technical problems tends to be greater, therefore stimulating questions/discus-

---

Lieutenant Colonel William G. Fohrman, USAF, is Director of Configuration and Data Management, Deputy for Acquisition Support, Aeronautical Systems Division, Air Force Systems Command. In 1974 he participated in the Air Force Education with Industry program as Manager of Research, Chemical Systems Division, United Technologies Corporation. Lieutenant Colonel Fohrman is a graduate of the United States Naval Academy and holds an M.P.S. degree from Golden Gate University.

sions that lead to better configuration management performance. There are also many intangible benefits to be gained from this arrangement. While I do not advocate that every configuration manager be an engineer, I do feel that configuration managers should have a good technical background, and that these people should be supported by clerical or administrative people, or by configuration assistants at appropriate grade levels.

With a base of technical configuration management personnel, we could better address the technical management issue. While the program manager is ultimately responsible for all aspects of a program, he generally tends to turn to engineering for all things technical, including technical management. The result in many cases is that engineering is expected to provide configuration (technical) management as well as the primary function of ensuring item performance. This dilution does not benefit either discipline, is counter to engineering's primary purpose, and contributes to repeated problems in the program development cycle.

Engineering generally writes the system specification, while all parties contribute to the statement of work. The project manager, particularly if he is responsible for multiple programs, may have neither time nor the technical qualifications to make a thorough review of these two documents to ensure that they are reflective of each other; that they are integrated from a technical management viewpoint; or that they adequately communicate to the potential contractor the tasks to be done. Such technical management assessment from the configuration manager, together with the performance assessment from engineering, could allow the project manager more time to devote to total program considerations.

Configuration management has an impact at each milestone of a program. The scope of this involvement is not necessarily limited to the following questions and the considerations of interest in the configuration manager's process.

To simplify illustration, the following program event or milestone schedule specifically applies to small programs. Large programs incorporate the same basic elements but the sequence of certain milestones may vary or be repeated.

- Receipt of Form 56 (PMD).
- Pre-program strategy meeting—The configuration manager must be a team member to review and evaluate: scope of effort, program direction, configuration management, manpower availability, estimated manpower expenditure, and program milestone structure.
- System specification (tech exhibit)—Configuration management should review and evaluate issues by asking: Is it technically manageable? Is it integrated with another project? Does it flow logically? Are interfaces stated? Is it limited to statements about performance of the equipment? Are requirements for quality and testing included? Is it sufficient, together with the

statement of work, to drive the contractor development specifications?

- Statement of work (SOW)—Configuration management input and coordination (Section J of contract, if no SOW) might include the questions: Are performance criteria reflective of configuration management? Is documentation addressed? Are schedules addressed? Are criteria stated, aside from MIL-STD-1521A, that clearly convey to the contractor what constitutes successful completion of PDR, CDR, FCA, PCA, etc.?
- Configuration management plan and program management plan—Configuration management considerations include: Define how to handle ECPs, define how to handle interfaces, and provide construction consistent with AFR 800-3.
- Purchase request (PR)—Configuration management must ask: Does PR include elements that impact configuration management requirements?
- Pre-RFP Data Review—Consolidate configuration management requirements and review strategy and input prior to release of RFP.
- Post-RFP review—Consolidate configuration management requirements and: assess reasonableness of contractor's submittal; establish negotiating priorities; and review contractor's preliminary configuration management plan.
- Pre-contract negotiation—Assure: clear statement of *requirements*; clear milestone definitions; establish functional baseline; and require ECPs for future functional baseline changes.
- Post-contract review—Ensure contractor counterpart has a common understanding of the contract; and ensure contractor understands EC requirement for changes to system specification.
- Preliminary design review (PDR) (may require iteration)—Approve contractor's configuration management plan; approve contractor's drawing structure (contractor should have completed most of the conceptual drawings, guide at 80 percent); approve development specifications (are they in concert with systems specification?); establish allocated baseline (no later than CDR); and ECP required for future development specification changes.
- Critical design review (CDR)—Approve update of configuration management plan; production drawings should be well along (guide at 80 percent); and review draft of product specification (track with development specifications?).
- Functional audit—In most cases this audit can incorporate the FQR: Accomplished after qualification test and qualification test report in most cases; and review documentation to ensure that development specification, product specification draft, drawings, and qualification test report are consistent.

- Physical audit—Approve product specification; approve final drawings; set product baseline; and ECP required for future product baseline changes.
- PMRT—Participate as appropriate.

Small programs are often rushed to contract because of financial considerations, give the illusion of simplicity, and often have less experienced project managers. For any program, however, configuration management is in a position to influence the conduct of the program in the critical early phases, and to make a positive contribution to the management effort. It is normal practice for program management to turn to engineering for performance support, to turn to the comptroller for financial management support, and to turn to program control and procurement for their respective support.

Configuration management is in a position to contribute a great deal more than administrative and clerical functions. Let program managers turn to configuration managers for configuration *management*. The AFR 65-3 is under review to incorporate this philosophy.

The support of industry and government in illuminating the configuration management discipline and its contributions is essential to more effectively "manage" the configuration of new systems and equipment. ‖

59 In-Progress Checklist Reviews for CM Systems

June Wohlgethan

This paper proposes a simple technique that can help a company ensure the adequacy of its basic configuration management (CM) support systems and at the same time impart insight into the broad spectrum of CM interests and methodologies necessary to meet contractual requirements. It suggests a relaxed and fundamental approach to CM through planned question-and-answer sessions between customer and contractor. The idea was developed to supplement AFCMDR178-1, Contractor Management System Evaluation Program (CMSEP), which is based on the principle that product quality is a direct result of management quality, and on the assumption that a responsible contractor, dedicated to the delivery of a quality product within cost and on schedule, will develop a management system in an orderly and planned manner.

Briefly, the recommendation is to incorporate into the government inventory of military standards a document defining "in-progress checklist reviews for configuration management systems." If selected for contract application, the checklist reviews would be planned to enhance program needs and be conducted in parallel with scheduled design and program reviews. The proposed standard would consist primarily of checklist questionnaires directed at various CM-related activities. A discussion of these in-progress reviews, with sample checklists, is presented herein.

The Importance of In-Progress Reviews

In-progress checklist reviews for configuration management systems would be valuable for several reasons. First, they would help communicate CM interest and convey basic CM systems to individuals from engineering, purchasing, manufacturing, quality, and logistics environments who are in some way involved in the tasks of identifying the physical characteristics of a configuration item (CI), controlling changes to those characteristics, and recording and reporting change processing and implementation status. These individuals, in spite of their contributions to the CM network, often have limited knowledge of CM and its functional interface activities pertaining to identification numbers and marking; configuration specifications; drawings; engineering document release; standards; change control; baseline control; manufacturing controls; logistics support controls; subcontractor controls; configuration verification; configuration audits and reviews; configuration status accounting and reporting; field support operations; and interface control.

June Wohlgethan is Head of Configuration and Data Management Systems and Audit at Hughes Aircraft Company, Microelectronic Systems Division, Irvine, California. As such, she provides training and consulting services to the division and is responsible for all configuration and data management customer audit and interface activities. As a result of 17 years of association with military and commercial customers, she advocates an open-door policy between customer and company and sees this policy as a key element to successful and cost-effective configuration and data management.

Secondly, in-progress reviews would promote early recognition of required CM program interfaces. Because reviews will commence early in a program cycle and may be repeated as often as desired throughout the various program phases, it would be possible to identify the need for accelerated or decreasing control levels. Also, because checklist questionnaires are directed to individuals actually responsible for task performance, the "players" are soon identified, communications opened, and priorities established.

The value of checklist reviews is also based on the fact that, subsequent to the pre-contract award survey, the follow-on contract statement of work, and early program reviews imposed by MIL-STD-1521,* CM may not be specifically addressed again by the customer until the time of the physical configuration audit, normally at the end of the development phase. A possible exception is the CM plan, which is often required with the proposal response and then again shortly after contract award.

In any event, pre-contract award surveys, statements of work, early reviews, and CM plans do not provide assurance that minimum CM controls will be initiated in a timely manner and maintained as necessary to support the technical objectives of both customer and contractor. That assurance can only result from the level of contractor understanding of CM activities, and this, in turn, can be influenced by the level of expressed customer concern. How to express this concern with no significant cost impact is the key to the proposed in-progress checklist reviews for configuration management systems.

The Checklists

In-progress checklist reviews would be conducted concomitant with contractual reviews; such as the systems requirements review (SRR), systems design review (SDR), preliminary design review (PDR), and critical design review (CDR). They may also be held during program reviews.

Selecting checklist(s) for contract application and identifying appropriate milestone event(s) for initiating and continuing in-progress reviews would be largely dependent upon contract requirements, the sophistication of a contractor's CM operations, and the confidence level achieved from past experiences. Checklist samples are presented in Tables I through V.

Questions Communicate at Working Level

Checklist questions, although simply stated, attempt to communicate specific types of controls valid for program application. They also permit system visibility. For example, when dealing with the subject of contract (baseline) specifica-

*MIL-STD-1521, Technical Reviews and Audits for Systems, Equipments, and Computer Programs.

tions, the question, "Who maintains the specification number assignment log?" assumes that such a log exists, indicates a one-person control point, and, when desired, opens the door for evaluating the system used for assignment of specification identification numbers.

During actual customer conduct of in-progress reviews, questions and related discussions would be directed to the working-level individual(s) responsible for a specific checklist item. Objective evidence of response would be verified. Questions not applicable to a specific review would be addressed in terms of pending plans and schedules, and those not applicable at all would be so noted. A sample format for checklist presentation is provided in Figure 1.

Whenever indicated by the nature of a procurement, checklist questionnaires are to be imposed on sub-tier contractors to ensure controls and further communications at this critical level.

Summary

Because configuration management is linked with company-wide activities, an increased awareness of fundamental CM control systems will serve to enhance communications between CM and interfacing departments. One way to achieve this visibility is through the in-progress checklist reviews for configuration management systems discussed above. These are customer-initiated question-and-answer sessions that provide a way to sample program CM control systems without disrupting normal work flow. Checklist questionnaires become working tools that provide guidance to non-CM specialists and enable product management and supporting functional organizations to meet contract CM requirements without large groups of dedicated specialists.

Checklists contain direct, fundamental questions that help to build CM concepts into product-oriented areas. They are designed to offset problems that could impact design, development, procurement, manufacture, test, operation, and maintenance activities. For example, a system that assures drawing and specification integrity also assures product integrity. Although not necessarily optimum in their coverage of a specific subject, the checklist questionnaires presented herein can be used as samples or guides in developing this concept more completely.

TABLE I
Checklist

Subject: **Contractual (Baseline) Specifications, Verify Support Systems for**
Program: ____________________

Definition: System, CI performance and CI design specifications are customer-prepared and/or customer-controlled documents that establish contractual baselines for program requirements, design requirements, and production configuration.

Objective: Ensure timely coordination, approval, tracking, traceability, statusing, availability, and baseline visibility of specifications and changes thereto.

1. Who coordinates customer specification review and approval/disapproval activities? Are status logs maintained?
2. Who maintains a current program specification tree (or equivalent) to show specification hierarchy, sub-tier specifications, and revision status? Is a current distribution list also maintained?
3. Who maintains specification number assignment log(s)?
4. Is there an active (on-call) specification review board for internal review and approval/disapproval of contractor-prepared specifications?
5. Has a specification review board membership and responsibility matrix (or equivalent) been published? Which members participate actively?
6. Where are specification review board agendas and minutes filed?
7. Who coordinates customer review and approval/disapproval of proposed changes* impacting customer-controlled specifications?
8. Is there an active (on-call) configuration control board for review and approval/disapproval of proposed specification changes?
9. Has a configuration control board membership and responsibility matrix (or equivalent) been published? Do all members participate actively?
10. Where are configuration control board agendas and minutes filed?
11. Who impounds and controls approved specification originals and approved SCNs, NORs, or equivalent? Are these documents formally released through an engineering release group?
12. Who incorporates approved changes into approved specification originals?
13. Who checks specifications to verify change incorporation? Are check logs maintained?
14. What is the turn-around time (from point of final approval) for specification change paper release and distribution?
15. Who maintains specification history files?

*Changes include engineering change proposal (ECP) with specification change notice(s) (SCN), no impact change memo (NICM) with SCN(s), notice of revision (NOR), and deviations.

TABLE II

Checklist

Subject: **Engineering Drawings, Verify Support Systems for**

Program: ______

Definition: Engineering drawings are the pictorial, tabular, and narrative descriptions of portions of CIs being developed or manufactured.

Objective: Ensure drawing approval, availability, traceability, baseline visibility, and change control.

1. Who is responsible for drawing preparation?
2. Is a current drafting manual in use? List title and date of issue.
3. Have contractual requirements for drawing preparation been published? To whom were they issued?
4. Who maintains drawing number assignment logs?
5. Have drawing approval requirements been published?
6. How are drawings coordinated for review and sign-off?
7. Who is responsible for change* paper preparation?
8. Is there an active change review board for review and approval/disapproval of changes not impacting a customer baseline?
9. Has a change review board membership and responsibility matrix (or equivalent) been issued? Which members participate actively?
10. Is there an active configuration control board for internal review and approval/disapproval of proposed changes impacting a customer baseline?
11. Has a configuration control board membership and responsibility matrix (or equivalent) been issued? Which members participate actively?
12. Where are configuration control board agendas and minutes filed?
13. Who maintains number assignment logs for change paper identification?
14. Who incorporates approved changes into drawing originals?
15. Who checks drawings to verify change incorporation? Who checks narrative-type drawings?
16. Who is responsible for engineering release, impound, and control of approved drawings, and change paper? Is release formal or informal?
17. Who is qualified to remove drawing originals from "impound," and on what authority?
18. What is the turn-around time (from point of final approval) for drawing/change paper release and distribution? Is a current distribution list maintained?

19. Where are the subcontractor/vendor drawing files?
20. How are subcontractor/vendor configuration baseline records maintained? How are baselines verified?
21. Does the drawing tree (or equivalent) uniquely identify: specification control drawings; source control drawings; altered item drawings? Who maintains this information?

*Engineering change proposal (ECP), engineering change request (ECR), engineering change order (ECO), engineering order (EO), etc.

TABLE III
Checklist

Subject: **Test Procedures, Verify Support Systems for**

Program: ______________________________

Definition: Test procedures establish detailed test objectives, test preparation, test equipment configuration, test precautions, non-test handling operations, and step-by-step operations for obtaining data to be recorded within stated tolerances and accuracies using a specified equipment configuration.

Objective: Ensure timely coordination, approvals, tracking, traceability, statusing, availability, and baseline visibility of test procedures and changes thereto.

1. Who coordinates customer test procedure review and approval/disapproval activities? Is a status log maintained?
2. Who maintains the test procedure number assignment log?.
3. How are test procedures reviewed internally for approval/disapproval?
4. Have internal test procedure approval requirements been published?
5. What type of change paper is used to propose changes to test procedures? To implement changes?
6. How are proposed changes to approved test procedures reviewed and approved/disapproved internally?
7. Has a test procedure review team membership and responsibility matrix (or equivalent) been published?
8. Who coordinates test procedure change review and approval/disapproval with the customer? Is a status log maintained?
9. Who impounds and controls approved test procedure originals and approved change paper?
10. Who incorporates approved changes into test procedure masters?

11. Who checks test procedures to verify change incorporation? Is a check log maintained?
12. What is the turn-around time (from point of final approval) for test procedure/change paper release and distribution? Is a current distribution list maintained?
13. Who maintains test procedure/change paper history files?
14. Who generates the equipment configuration baseline status report for equipment entering test? Where are report history records maintained?
15. How is needed test data requested from subcontractors/vendors? Where is it stored?

TABLE IV
Checklist

Subject: **Change Effectivity Integrity, Verify Support Systems for**

Program: ____________________

Definition: Change effectivity is that point in the manufacture of a part at which an engineering design change is introduced. "CI change effectivity" applies to all CIs scheduled for manufacture, including spares, and is expressed in terms of the part and serial numbers of the CI. When a change affects an item below the CI level, the term "production effectivity" is used. Changes at the production effectivity level look to successive next-higher assemblies until the first serialized CI is reached so that the change can be communicated to the customer at the CI level.

However, in support of contractor material and manufacturing planning needs, production effectivity points are expressed on a firm or estimated basis and established in light of the contractor's processing capability and the customer's change approval reaction time, as well as on engineering procurement and manufacturing lead time considerations.

Objective: Ensure that customer CI effectivities are stated correctly, and that correct input is received from engineering and manufacturing in order that production effectivity points may be targeted and implemented at minimum cost.

1. Who has the ultimate responsibility for assuring that correct change effectivity is specified on proposed/approved engineering change paper? Is this accomplished at change board?
2. Who is the quality engineer responsible for verifying that serial number effectivities of engineering equipment changes agree with the change board actions that approved/authorized the change? Who is the manufacturing

engineer with this responsibility?

3. Who ensures timely delivery of the approved engineering change paper to the responsible manufacturing and quality engineers?
4. Who is responsible for assuring that effectivities recorded on manufacturing planning paper are the same as those on approved engineering change paper?
5. Who determines what point in the manufacturing cycle to introduce an approved engineering change? What considerations are used?
6. Who prepares revised manufacturing planning or adds a revision sheet to the original planning for each serial numbered item affected by the engineering change?
7. How is the manufacturing planning revision sheet identified? Is the engineering change number added to the original planning paper?
8. Who is the quality engineer responsible for placing an approval stamp next to completed manufacturing planning inspection points to authorize continuation of manufacturing operations?
9. Who maintains serial number logs and records? Who verifies these records?
10. Who maintains lot number logs and records? Who verifies these records?

TABLE V
Checklist

Subject: **Traceability, Verify Support Systems for**

Program: ______________________

Definition: Traceability is the characteristic of a product and its parts that enables the parts to be tracked down to their source of origin and date of manufacture. Traceability applies to parts and to engineering, manufacturing, and test documentation as well.

Objective: Ensure that each supplier of CI parts, as well as the company, keeps identification records for each part. Then when a product failure occurs, and the part that failed is analyzed and found to be an inherently defective part, it can be traced to its source. At that time, the defective part can be identified, traced to its current use or application and, if necessary, replaced with a more reliable part.

1. Who ensures that procurement, inspection, quality assurance, manufacturing, and engineering are following the system established at the beginning of the project for recording the origin and present location of all parts, materials, modules, and subassemblies?

2. Who is responsible for preparing and maintaining company records for each item manufactured that provide the date of manufacture, date of final acceptance, test data, part failures or defects, and identification numbers?
3. Are suppliers of CI parts tasked to maintain records for each item delivered that provide the date of manufacture, date of receipt by the company, test data, part failures or defects, and identification numbers? Is this accomplished through a formal purchase agreement?
4. Who prepares part or kit lists (material control documents) that identify each part that goes into a subassembly or assembly?
5. Who records part identification in the form of serial or material numbers on the part or kit list?
6. Who ensures that the part or kit list and manufacturing planning paper, which describes the parts installation and assembly procedure, accompanies the part through final assembly?
7. Who records on inspection all data pertaining to lot-numbered parts such as test results, inspection, and manufacturing records?
8. Who maintains the room for storing parts? Is it a controlled area?
9. Who maintains logs of parts received and issued by the storeroom?
10. Are all parts inspected for damage, quantity, and Q.A. requirements and approved by the inspector before being accepted by the storeroom?
11. Who assembles storeroom parts into kits, based on approved kit lists?
12. Who is responsible for storeroom inventory control? Is a report published?

Control of Documentation: Avenues to Failure

68

Carl P. Hershfield

Most industrial concerns today—and especially those containing manufacturing facilities—maintain, as vital to the overall success of the operations, some measure of control over their engineering data. The degree of control varies with the type and size of the company, the general rule being that the smaller organizations have less need and therefore tend to exercise less rigid and more informal controls over their data, while the large production organizations maintain a rigid, though not inflexible, grip on all data that has been issued or released for manufacture.

Most businesses, however, lie somewhere between these two extremes and contain within the framework of their service organizations a functional group whose duties include, but are not necessarily limited to, the issuing, recording, reproduction, distribution, storage and, subsequently, the exercise of change control over engineering data that have been properly authorized for use by the originators. The group may be known as "documentation control," "drawing release and control," "data control," or some similar name implying the control of documentation. Whatever you call it, this group is the funnel through which all data flows in a formal, organized manner between the engineering and manufacturing groups—or between engineering groups themselves. It is to the people who generate and supply the information to be transferred that this paper is directed.

Suggested Approaches

Outlined herein are a few of the many ways that have been tried with varying degrees of success to shortcut the controls in the interest of expediency. Other methods not mentioned are either too amateurish for our consideration or are too easily detected. Therefore, I will outline only those methods which, from my own experience, have worked extremely well—for a time.

A proper attitude is mandatory if you are to successfully employ these procedures. First and foremost, you must believe that sound and permanent results can be sacrificed to expediency. If you cannot cling to this principle, you may as well give up and process your data in the manner prescribed by your company and customer.

Carl P. Hershfield is Manager of the Configuration Management Department, Communication Systems Division, GTE Sylvania. He has held a number of management positions on major defense programs in the areas of engineering support and logistics management. He has been a visiting instructor at the Defense Weapon Systems Management Center of AFIT, and a thesis advisor to M.B.A. candidates at Boston College. He holds a B.S.M.E. degree from Worcester Polytechnic Institute and is a registered professional engineer in the Commonwealth of Massachusetts.

Second, engineers must reject the philosophy that the primary output of engineering is pieces of paper. Everyone knows that hardware and software are more important than documentation, and that manufacturing can't produce the final product without *some* direct assistance from engineering.

Third, engineering types must believe that administrative procedures and controls cannot logically be applied to them because of the exacting, delicate nature of their work. Engineering brainpower cannot be harnessed, scheduled, and controlled, if anything creative is to come of it. Really convince yourself of this.

Fourth, consider the time-worn definition of the distinction between scientists, managers, and engineers:

> A scientist discovers what can be done; a manager decides whether or not to do it; an engineer decides how to do it economically.

If you number among the managers, don't assume that the above definition is as limiting as it seems. A good manager certainly will pitch in and make engineering decisions—or whatever else needs to be done—in order to get the job out. Engineers, too, should follow this example and not be afraid of getting their hands dirty. A little personal expediting here and there can really break bottlenecks and get things moving. More about this later.

Fifth and finally, if you don't already have one, develop a haughty disdain for peripheral operations—control groups in particular. They only slow things up through their red tape and make no tangible contribution whatever to the success of the project.

Shall we now take a detailed look at some of the ways in which you can bypass this insidious bottleneck and perform in a much more efficient and uninhibited manner? If you have sufficiently embraced the foregoing tenets, you should next consider which of the three basic avenues of approach you should select.

APPROACH A

Convince management that the procedures are not adaptable to your project and that an attempt to implement them will seriously jeopardize the outcome. Cite specific areas where you can't possibly be expected to conform to such paper procedures and still manage to have equipment standing when it will be needed. Lean heavily on the cost angle and the certainty of schedule slippage. This is sure to shake them up. When you have them convinced that the normal procedures and controls should *not* apply to *your* project, have someone with authority issue an edict to this effect. If you happen to be a VIP, issue it yourself. (Make sure copies go to the control and procedures people.)

APPROACH B

Study the procedures diligently. Take them home with you and get to know

them backwards and forwards. If you concentrate entirely on what they say instead of what they mean, you should be able to pinpoint all the loopholes and inconsistencies. This will enable you to sidestep the routine in an intelligent manner and will also provide you with an excuse if you are found out.

APPROACH C

If you haven't already read the procedures—don't. Unless you are really interested enough in this sort of thing to take Approach B, they'll only confuse you. It will help if you don't even bother to find out what procedures exist to govern your activity. If anyone complains that you are not conforming to the rules, assert your ignorance. Say you never received a copy.

As a guide for selection, Approach A is especially suitable for managers and project engineers and others with solid experience. A thorough understanding of engineering procedures is a prerequisite. It's a hard-sell approach and should never be attempted by neophytes. The main advantages are that it "wipes out" the service functions in one fell swoop, is entirely above board, and adds immeasurably to your status image.

The second approach, B, is more adaptable to those with limited authority but major responsibilities. This approach requires a high level of intelligence and if applied properly should enable you to get through the peak phase of your project before any repercussions are felt. It is hard work, but is highly recommended.

Approach C is the only doctrine left to embrace if you are unable or unwilling to take either of the more direct approaches. It's the easiest way and a good approach to take if you are not particularly interested in whether your data is controlled or not. If you *are* intent on circumventing procedures, however, this approach is not recommended, as it is rather easily detected by control types who have nothing else to do but look for such things.

If you have taken Approach A and have been successful, you have it made. The way is open for you to operate in any manner you deem appropriate. Adopt the attitude that procedures are like company standards—handy to have around if you want to use them, but not mandatory. Select those portions of the procedures that suit your purpose and employ them. Ignore the rest. If anyone complains, simply flash the edict and the matter is ended. For those who elect to take either of the other approaches, the following detailed procedures are recommended as highly workable; although you may have to modify them slightly to suit your particular needs. This is intended to be an heuristic, thought-provoking list of ideas which, if studied, will surely spawn some new and original procedures to plague the control personnel.

PROCEDURE 1

Make a big fuss about not having time to waste reading procedures. With a little effort, you should be able to convince your boss that taking that much time

out would really have fatal consequences. Be aloof. Show by your demeanor and sense of urgency for the job that you are above such detailed interests. If he's really insistent, make it clear that you don't consider attention to such detail to be the responsibility of a creative person and suggest he hire a couple of expediters to coordinate the information. Chances are the budget is too tight to allow this and he'll drop the whole matter.

PROCEDURE 2

Don't worry too much about details or good drawings when you are trying to get out a release. Make minimum drawings, release sketches, and layouts; plan on a redraw program later on. (The chances are that you will be on a new project when that time arrives, and you won't have to worry about it.) If manufacturing has difficulty understanding the drawings or picks up errors, they're sure to call you, and you can then give them the correct information. You can always pick up the errors later with change notices (if you don't forget). The important thing is to meet your scheduled release date so that you can make a good progress report next month.

PROCEDURE 3

Enlist an accomplice in production engineering or production control. After all, these people are anxious to get information as quickly as possible; and if proper overtures are made, you should be able to convince someone that the successful outcome of the job depends on a "close" working relationship. This is an important step—and you should make every effort to establish this rapport because the success of several of your operational methods will depend on your having this unquestioning co-worker. Choose the supervisor if at all possible.

PROCEDURE 4

Make preliminary prints of drawings and data, and hand carry them to manufacturing so they won't have to wait for officially released prints. If anyone questions it, tell him that the data are actually in the process of release at that moment and will be there later that day or the next morning for sure. You are merely "helping out" by letting them get a head start. When you get back to your own area, don't worry too much about urging your people to get the data to the release group; you already have the manufacturing wheels turning.

PROCEDURE 5

When you *have* to write change notices, be are brief as possible. Simple, clear-cut statements such as "Change per marked-up-print" serve as red flags that changes are pending and enable you to satisfy the requirement for advance notification without consuming much of your valuable time. Of course, this doesn't give manufacturing any indication of what is being revised, and they will

have to wait until the drawings are actually changed and new prints issued before they can do anything, but they will have had advance information. (Warn the draftsman not to lose the marked-up print!) This is also an excellent means of causing manufacturing to cease work on a piece when you don't know exactly what changes you are going to make. It's practically as good as a stop-work order and not nearly as embarrassing for you.

PROCEDURE 6

A good way to make quick changes is to run off bootleg copies of your change notices before you submit them to the change control group. You can hand carry these to manufacturing immediately, and with a little show of urgency convince them that you are doing them a huge favor in supplying such "hot" information, even though the documents may not have control numbers or full complements of authoritative signatures.

PROCEDURE 7

The type and number of changes forthcoming on a project may be used as a subtle yardstick in the measurement of managerial control exercised on that project. If you can manage to keep the responsibility for correcting released drawings within your own group, you can cut down tremendously on the number of reportable changes you'll have to write. When you have a drawing out for revision, you can also incorporate all those "incidental" corrections for which you never got around to writing notices. If you change anything really important though, better call manufacturing and tip them off—because there's absolutely no way for them to know what was changed, unless they make a line-for-line check of the new print against the old.

PROCEDURE 8

A more sophisticated method for effecting quick changes is to maintain a second set of reproducibles of your data (photo-tracings are pretty good). In this way you can make changes to the second reproducible after the released original is locked up in the control files. New prints can be delivered into the shop without any red tape whatsoever. You recognize that these aren't bona fide changes, of course, so don't record the changes or raise the revision symbols on the second reproducible. Plan to pick up all such changes the next time the controlled original drawing is out on change. (It will save you a lot of grief if you can manage to do this before the altered part reaches inspection, because you now have two or more prints in the shop with the same revision status but containing different information—and if the part should be inspected to the wrong print, it could be rejected.)

PROCEDURE 9

Be original. Devise little procedures and forms of your own. If you top this off with your own private numbering system, you really have a neat little scheme that works like a charm and keeps everyone else in the dark. Private systems have the advantage that no one else can operate them if you happen to be home sick or on vacation. They'll have to wait until you get back—a fact which adds to your prestige through the implication that no one can do the job but you.

PROCEDURE 10

Finally, make use of every opportunity to "do-it-yourself." "Help out" manufacturing by appearing in person regularly to issue oral instructions, mark-up-prints, etc. Make the rounds daily and do a bit of expediting here and there. It really pays off if you visit incoming inspection on each trip and waive inspection on parts that are held up for any reason. Nine out of ten, they'll do the job O.K.! If you've had sufficient foresight, you will have procured many of the critical, long-lead items yourself (preferably through petty cash) and thus will have assured that they are on hand when needed. There won't be any procurement or inspection records, certificates of compliance, etc., but you certainly will have saved a lot of time and trouble.

Remember—expediency! That's the important word. And, oh yes—there are a few things that perhaps should be mentioned. All of the aforementioned laudable endeavors make it horribly difficult, if not downright impossible:

- For drafting to produce a set of drawings that accurately reflect the equipment as produced and that meet contractual requirements for deliverable drawings, without recourse to a complete re-draw;
- For the service organizations to gather the records necessary for the procurement of vendor data—often a contractual commitment;
- For manuals to be written;
- For spares to be provisioned;
- For quality control and inspection records to be supplied;
- For the equipment to be reproduced;
- For the myriad other things to be done that are required and that must be completed before you can call the job "done" and get paid for it.

Unfortuntely, I do not have the solution to offer for this enigma if you choose to use the short cuts. And, unfortunately, I do not know of anyone who does.

Do you?

Configuration Management and Software Development Techniques

74

Zygmunt Jelinski

The need for configuration management in software development is directly related to the technological advances made in recent years in both software and hardware. Large-scale computers are larger, faster, and cheaper than they were 10 or 20 years ago. Their physical size, through the microminiaturization of components, has become many times smaller. Computer cycle time is now measured in nanoseconds rather than micro or milliseconds. The availability of the peripheral equipment, such as disks and magnetic tapes, helped to increase the memory capacity, at a reduced cost. For special applications, minicomputers have been marketed in all types and sizes. With the progress of aerospace technology, microprocessors are being used to replace many human and mechanical functions. With the use of telephone and telegraph lines, transmission of information between remotely based computers has become commonplace. Computer networks have been developed and connected by software systems. Time sharing and the use of computer terminals are the way of life in many organizations. In software development, new techniques are being applied to software production, checkout, and test.

In large military programs, more and more attention is being focused on software life cycle costs, because 75 percent of software maintenance and modification costs occur after software becomes operational. This is where the importance of configuration management cannot be overemphasized, especially since configuration control and status accounting must be implemented throughout the life cycle. When, and to what degree, configuration management should be implemented in the software development cycle depends on the requirements of the program. Traditionally, software development in a system goes through a number of stages, i.e., requirements, definition, design, coding and checkout, testing and integration. At some stage, configuration management should be initiated. In this paper I intend to review some of the tools and techniques that have been developed to make each stage of development more comprehensive and more efficient, and to provide a better understanding of the software problems to project and configuration managers. Specific disciplines are being applied to each stage of software development in order to minimize the cost of software and the incidence of errors.

System Requirements, Specifications, and Design

Some definite steps are now being taken to ensure that system requirements are well defined and well documented. Every government request for proposal

Zygmunt Jelinski is Chief of the Computer Sciences Branch of McDonnell Douglas Astronautics Company. He directs research in software reliability, language processing including meta-assemblers and meta-translators, simulations, and emulations. Mr. Jelinski holds both B.S. and M.S. degrees in mathematics from the University of London.

(RFP) for a new system calls for a thorough analysis of system requirements, and computer languages are being developed to assist requirements formulation. These steps provide a basis for good software design and are conditioned to minimize design errors. Studies have been conducted at NASA and in the Air Force to develop a requirements language, and to make the task of specifying requirements easier and more precise.

System specifications and the manner in which they have to be documented have long been outlined in, among others, MIL-STD-490, MIL-STD-83490 and SECNAVINST 3560.1. Specification languages have been developed to assist specification writers and software analysts in outlining systems specifications. One example is computer requirement analysis (CRA), promoted by the Air Force, which is the extension of program specification language (PSL) and program specification analysis (PSA) developed by Dr. Daniel Teichroew.[1] System design errors are further minimized by using other techniques such as program design languages (PDL), assertion methodology, and by proving program testing instrumentation and testing at the design level, and then extending this technique in detail to the lower levels of design. Proving program correctness is in too early a stage of research to be an effective tool of design error reduction.

Computer Program Management Techniques

In recent years, it has become obvious that many errors in software occur because of the lack of good management of software projects. In response to this problem, two significant techniques, structured programming and chief programmer teams, have been put into practice. Structured programming was originally advocated by E. W. Dijkstra, and was adopted quite enthusiastically in many aspects of the software development practices. Two main features of structured programming are the use of standard constructs and the improvement in the documentation of programming texts by use of indentation; and an increase in the number of comments. Such constructs as "sequence," "if-then-else," "do until," "do while," and "case have" are now standard in most structured programming applications. Top-down programming is another software management technique. According to Dijkstra, the basic approach in top-down programming is to compose a program in minute steps, deciding as little as possible at each step.[2]

The essence of top-down programming is the division of a large program into a number of smaller sub-programs (these sub-programs may be sub-routines, procedures, blocks, macros, etc., depending on the programming language being

1. Dr. D. Teichroew, ISDOS Project, Department of Industrial and Operations Engineering, University of Michigan, March 1975.
2. E. W. Dijkstra, "Notes on Structured Programming," Software Engineering Techniques, NATO Science Committee, 1969.

used). Dijkstra suggests that there are at least four ways of conceiving subprograms:

- Standard routines to be used as needed;
- Objects to be conceived and constructed by the user to reflect his analysis;
- A device for the reduction of program length;
- A means for rebuilding a given machine (computer) into a more suitable one.

The starting point is the program specification, which provides requirements for the program development process. The highest level requirements are addressed at program inception, whereas decisions on detailed specifications are delayed to the later stages of the development effort.

Much of the early stages of program design is usually stated via English language statements. Thus, early versions of the program will consist largely of computer instructions, interspersed with English language comments that document in detail the decisions that have already been made. Such a program is self-documenting to a very high degree. A running program is in existence virtually from program inception. Integration is accomplished by adding refinements to the existing program in the top-down manner. Program testing is also a continuous process performed throughout the development cycle.

The use of structured programming and top-down programming with self-documentation is proving effective in decreasing design errors. The concept of a chief programmer team was formalized by Harlan Mills of IBM, in the well-publicized *New York Times* experiment.[3] The nucleus of a chief programmer team is a chief programmer, a backup programmer, and a programming librarian. This nucleus is standardized to provide management continuity not only for programming expertise, but also for project recoding and documentation.

The chief programmer is a technical manager, to whom all team members report directly, whose principal job is to design and code top-level programs and to examine code written by other team members.

The backup programmer is a peer of the chief programmer in program design and development. He is involved in every aspect of the work and participates in making all important decisions. He can also provide test planning for the system independent of the chief programmer. He is an important member of the team, preserving the continuity of the project should the chief programmer leave or be reassigned.

A programming librarian is the team member who maintains the records of a project in the development support library in both internal (machine-readable) and external (human-readable) form. External records are the project records in human-readable form that are maintained in a set of file listings and that define

3. H. D. Mills, "Chief Programmers Teams: Principles and Procedures," IBM Federal Systems Division, 1971.

the current status as well as the history of the project. Current status is maintained in looseleaf notebooks, each headed by a directory and followed by an alphabetized list of member modules.

According to F. Terry Baker and Harlan D. Mills, chief programmer team operations provide increased productivity by sharply reducing the debugging and reworking required in a project.[4] The initial coding requires the same amount of time, but the design-level thinking is transmitted deeper into the coding by technical and organizational means. Structured programming displays program organization and interactions more effectively for the coding process. More competent, but fewer, people do the coding with carefully orchestrated teamwork. The results are increased productivity and, even more significant, improvements in the reliability and maintainability of the code produced.

There is no question that a management-team approach to software development is an improvement over the past practices where many a programmer was left to his own devices. It brings attention to the fact that the initial software development planning and design can be costly items, but the initial investment, if made, should pay off handsomely in increased software reliability.

Software Development Tools

Software development in the early days of computers was primarily applications oriented, and programs were written to solve a problem. Once the problem was solved and results produced, the program was abandoned. But it was discovered that many integral parts of a given problem repeated themselves and that the code for that problem could be re-used. The first software development technique was thus to save the code for possible use on another occasion. With an appropriate procedure having its own entry and exit, that code came to be known as "sub-routine." The next software development tool was the assembler. This was a program to translate symbolic code into machine code, and to provide appropriate diagnostics. As a result of the use of sub-routines and assemblers, fewer errors in software were encountered.

Diagnostic software was the next improvement in the software development process. Diagnostics, tracing routines, and debugging routines were used for checking out a new program. Input/output and conversion routines were used for the software building process. All of the applications software, at that time, were being developed at the machine-language or assembly-language level. The sub-routine libraries, however, provided the basis for building operation systems for large-scale computers. These became a standard delivery item.

4. F. T. Baker, "System Quality Through Structured Programming," *AFIPS Conference Proceedings*, 1972.

With the advent of higher-order languages (HOL), it was necessary to develop a software system that would interpret the HOL statements and generate machine language code. The resultant program could then be run on the computer. Such a software system was called the "compiler" and proved to be a major step in the software development process. Writing of a compiler was a lengthy and rather expensive process. Compilers had to be written for each computer, and their price tags, depending on capability and versatility, often went up to $1 million. Compilers were never transferable and differed in technique, as well as in capability, from computer to computer.

As the science of software development progresses, the tools themselves are being written in higher-order languages; for example, a compiler or an assembler is written using a meta-language, and a resultant program is based on a higher-order language. The advantage of this approach is the transferability between computers. One example of such transferability can be illustrated in the application of a meta-assembler. A meta-assembler can be written in a higher-order language such as FORTRAN and, in general, can be resident on a large scale (host) computer. Its function as a software development tool is to process programs written in any assembly language, and to produce object code for other (target) computers. Because assembly languages are usually computer-dependent, the input to the meta-assembler consists of both the syntax of an assembly language and the computer architecture of the target computer. For example, a meta-assembler written in FORTRAN and resident on a CDC 6600 computer may be capable of processing assembly programs for such minicomputers as the PDP-11 or Varian 620. The meta-assembler will then analyze the target computer assembly code and its architecture and output the target computer code.

An example of an input to the meta-assembler is shown in Figure 1. The advantage of having a meta-assembler as the assembly language program processor installed on a large-scale computer is that one does not have to write separate assemblers for each target computer. We can simply describe the computer characteristics, describe the assembly language, and have the programs processed in the language of that specific computer. This technique is fairly new, and with the proliferation of all types of computers, it tends to gain more acceptance as a software development tool because it is economical and avoids repetition of software-building.

A breakthrough came in the technique for language development when the Backus-Naur form was developed; the form allowed for the description of language syntax and some of its semantics.[5] The next step was to develop a processor that could process the notation and build a syntax analyzer for a given

5. Backus-Naur, *ALGOL 60 Development Conference Proceedings*, Paris, France, June 1959.

language. Such syntax analyzers have been accepted as a technique for compiler or language processor writing. This technique proved to be an advance in language processor development, and also contributed to the reduction of design errors. The sample of the meta-language description is shown in Figure 2.

FIGURE 1
Meta Assembler Example

SAMPLE INSTRUCTION SYNTAX/FORMAT

LDA GREG, ADDRESS, XREG

0 – 7	8 – 11	12 – 15	16 – 31
OP CODE	GREG	XREG	ADDRESS

STA GREG, DISP(XREG, BREG)

0 – 7	8 – 11	12 – 15	16 – 19	20 – 31
OP CODE	GREG	XREG	BREG	DISP

SAMPLE TARGET DEFINITION STATEMENTS

NAME	OPERATION	OPERANDS
	REGS	G=(0-15), B=(1-15), X=(1-15)
FORM1	IFORM	32, O=(0-7), G=(8-11), M=(L=(16-31)), X=(12-15)
FORM2	IFORM	32, O=(0-7), G=(8-11), M=(D=(20-31), X=(12-15), B=(16-19))
LDA	OPDEF	X'1A', FORM1
STA	OPDEF	X'3C', FORM2

The compiler writing system (CWS) is another software development tool geared toward automated compiler development of compilers. The need for compiler writing systems arises from the necessity to bring the computer closer to the user by providing a means of processing the application language. This development tool provides for multi-language processing. The target side of the CWS is directed to an individual computer on which that application is performed. The model of CWS is represented in Figure 3. The system itself will be written in higher-order language, or perhaps in a meta-language. Therefore, the system itself is host-portable, and is based on a large-scale computer.

FIGURE 2
Metalanguage Description

```
BACKUS-NAUR FORM SYNTAX DESCRIPTION

<SENTENCE>          ::=   <NOUN PHASE><VERB PHRASE>
<NOUN PHRASE>       ::=   <MODIFIER><NOUN>|<NOUN>
<MODIFIER>          ::=   THE | A
    .                      .
    .                      .
    .                      .

METALANGUAGE SYNTAX DESCRIPTION

$SENTENCE           .=    $NOUNPHRASE,$VERBPHRASE.
$NOUNPHRASE         .=    $MODIFIER,$NOUN//$NOUN.
$MODIFIER           .=    'THE'//'A'.
    .                      .
    .                      .
    .                      .
```

FIGURE 3
Model of Compiler Writing System

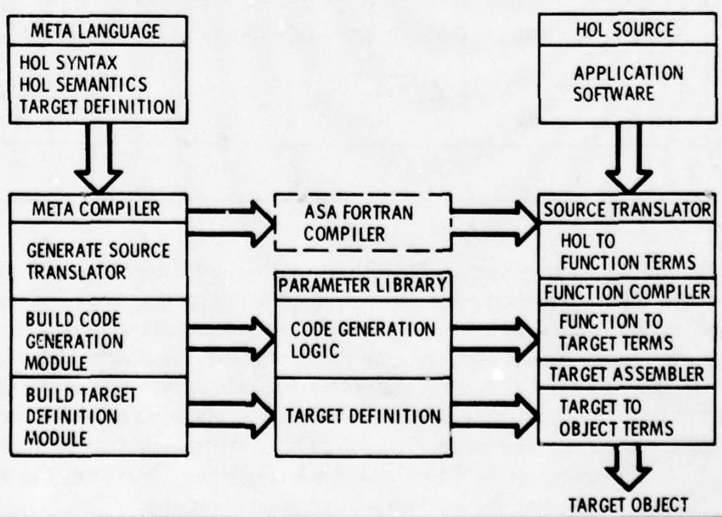
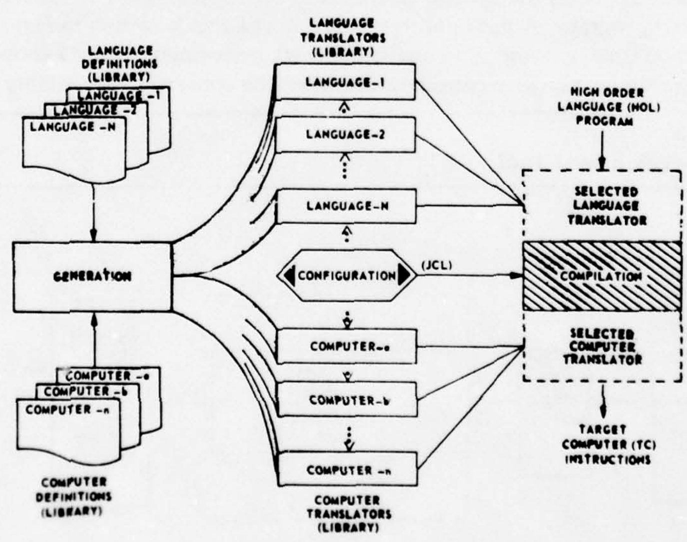


Figure 4 shows the schematic of a compiler-writing-system language and of compiler generation. The target computer architectures will be specified as one input, and the application languages definitions will be specified as another input. The CWS will then search the respective libraries and produce proper translators for the application language and the target computer. The translators will form a nucleus of the application language processor. The higher-order application language program is now ready to be processed. The process involves three stages: first, the source program is processed by the language translator; secondly, the program now represented by a CWS internal notation will be compiled and optimized; and, third, the optimized program will be merged with the target computer architecture, and the code will be generated to produce the target computer program.

An example of a language translator system is shown in Figure 5, where a test-oriented language (TOL) to the BASIC translator was developed. Both TOL and BASIC are higher-order languages, with TOL being application oriented. In order to translate TOL, it was necessary to build two processors. They were built with

FIGURE 4
CWS Language/Computer Generation, Configuration, and Compilation



the use of the meta-language. The first processor analyzes the TOL constructs and BASIC correlation functions. The syntax and semantics of the TOL language and BASIC correlation functions are formed into a dictionary. The dictionary provides the basis for language definition and language translation. The second module is a generalized processor that takes as its input TOL language programs and translates them into the BASIC code. The translation process uses the information stored in the dictionary.

The latest Department of Defense effort to develop one standard higher-order language—ADA for embedded computer systems—uses language translator techniques for ADA evaluation. Two competing contractors have designed their respective versions of ADA and have written HOL translators that translate computer programs into an intermediate language. Only one design and one contractor were selected by DOD to continue in the effort, although a number of industrial and academic organizations have been invited to participate in language evaluation and will use the available translator to process specific application programs written in ADA.

Software Validation and Verification

The problem of software quality is still a touchy subject in software development. In the past, the quality of software was seldom addressed before the actual construction process took place. It, in effect, constrained itself to the testing of a completed program. A pictorial representation of this is shown in Figure 6 with its effect on time, amount, and quality of work performed. Figure 7 shows the effect of continuous quality consideration when the concern about quality starts in

FIGURE 5
TOL Development Tool

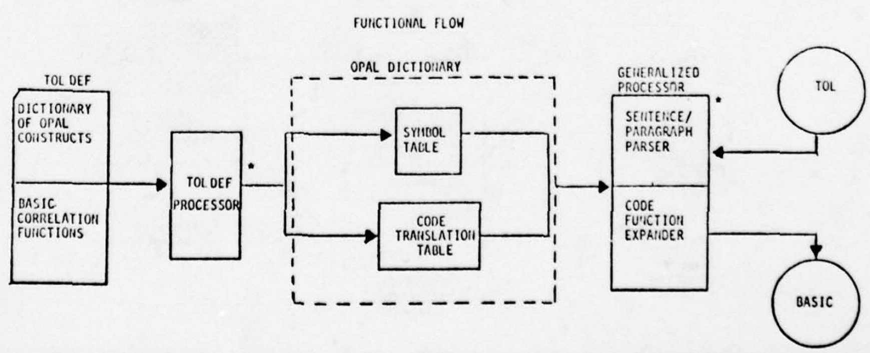
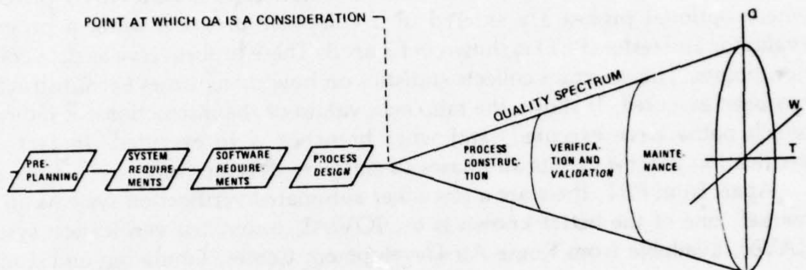
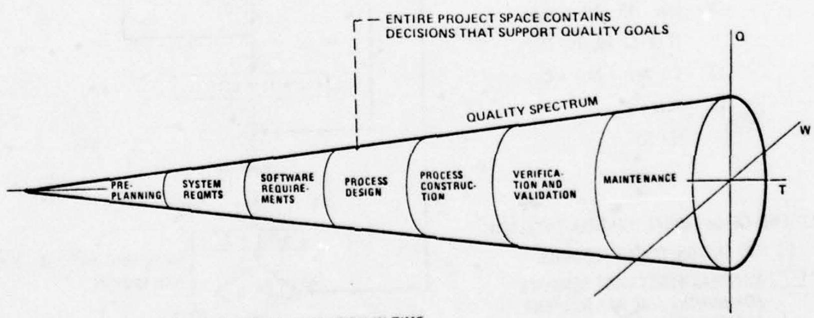


FIGURE 6
Effect of Delaying Quality Considerations



WHERE: T IS PROJECT DURATION IN TIME
W IS WORK PERFORMED
Q IS QUALITY EFFORT

FIGURE 7
Effect of Continuous Quality Consideration

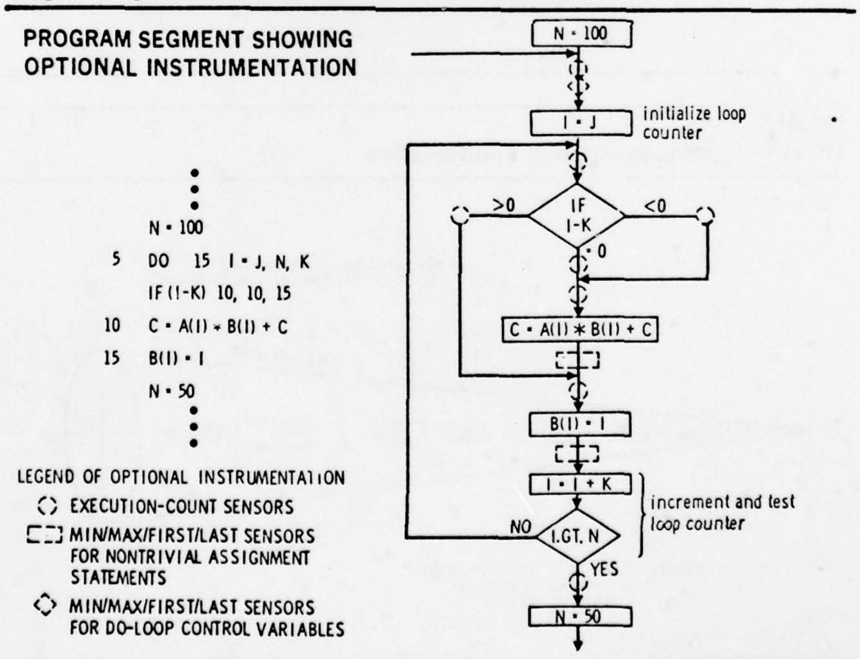


WHERE: T IS PROJECT DURATION IN TIME
W IS WORK PERFORMED
Q IS QUALITY EFFORT

the pre-planning stage and the beneficial effect of this concern is most apparent. There are a number of validation techniques that exist, but more and more emphasis has been placed on automated verification systems. The techniques that exist in the United States for the purpose of validation of software include not only the software instrumentation, but also other techniques, such as emulation and simulation. An example of the instrumentation of a FORTRAN program where optional probes are entered in a computer program using a program evaluator and tester (PET) is shown in Figure 8. These probes serve as data collection points. The program collects statistics on how many times each instruction has been executed. It shows the min/max values of the instructions; it indicates which paths were executed, and what branches were executed. In fact, the verification process affects all phases of software development.

Apart from PET, there are a few other automated verification systems on the market; one of the better known is the JOVIAL automated verification system (JAVS), available from Rome Air Development Center. Emulation and simula-

FIGURE 8
Program Segment Showing Optional Instrumentation



tion are also techniques that have gained acceptance as validation and verification schemes. For example, at MDAC, a QM-1 computer was used to develop a MARC I computer emulator to check out flight software. This technique of validation is useful because it applies the "software-first" principle, and allows for check out of software before a flight computer is available.

Simulation is also used for software validation. Since the advent of microprocessors, it has become standard practice to first simulate the instruction set and the environment, then use the simulator to check out application programs. The disadvantage of simulation lies in its cost where the slow-down factor between 100:1 to 1000:1 can make the process costly in computer time.

Support Software

With the advent of new software development techniques and the increased tendency to automate many development stages, support software is playing an ever-increasing role in the software community. The major reason for the existence of support software is to reduce the cost of software production, which has become a significant expense in production of computerized systems. Technology in computer hardware has been progressing and has resulted in microminiaturization, semi-conductor techniques, integrated circuits, and the research into very high-speed integrated circuits (VHSIC) which, in turn, make computer hardware costs almost insignificant; however, software production costs have increased for many reasons.

Some software costs are attributable to duplication, and some to unreliability. Higher-order languages such as FORTRAN, COBOL, PL1, ALGOL, CMS-2, and JOVIAL help to minimize the costs somewhat. Unfortunately, most of them are still machine-dependent and, therefore, the same program can rarely be run in an identical fashion on different computers. There is an organized attempt by DOD to restrict the number of higher-order languages, standardize them, and make them transferable between different computers. The development of ADA, and its introduction as one standard DOD HOL, will significantly improve the cost situation in 5 to 10 years. Because software systems are usually underfunded, adequate testing is not conducted; many software errors emerge after system delivery, adding to the unreliability factor. New programming techniques, such as structured programming, have helped to increase software reliability and keep the costs down. It is hoped that ADA will have a major impact on improving software reliability.

Support software can take many forms, starting with environmental simulators, language processors, assemblers, meta-assemblers, compilers, meta-compilers, compiler writing systems, automated verification systems, documentation aids, test case generators, and others.

The U.S. Navy defines software as all programs used in the development and maintenance of the delivered operational programs and test/maintenance programs.[6] Support programs include, but are not limited to:

- Compilers, assemblers, emulators, builders, and loaders required to generate machine code and to combine sub-programs or components into a complete computer program;
- Debugging programs;
- Stimulation and simulation programs used in acceptance test procedures;
- Training programs used in operator training sites;
- Data extraction and reduction programs applicable to operational programs;
- Test programs used in development of operational programs;
- Programs used for management control, configuration management, or document generation and control during development.[7]

Configuration Management of Support Software

One problem we should address is how to manage the configuration of deliverable support software. Should it be treated in the same manner as the embedded software, or should it be considered as an entirely separate item? Answers to these questions are difficult, but it is felt that each support software *program* should be considered on its own merits. For example, it is difficult to imagine that there will be changes in a deliverable documentation aid such as flowcharter. In most cases, it should be a standard "off-the-shelf" item and, therefore, configuration management should not be required. On the other hand, such programs as a compiler or a meta-assembler may require a lot of work before they can be accepted and delivered; therefore, configuration management control and the relevant procedures should be enforced.

Form of delivery and acceptance criteria also present problems; many software houses will deliver their software system in binary code in order to protect their source. This gives them the control of the program, and any future changes must be directed to them; on the other hand, source code delivery gives the control to the user. Acceptance criteria for deliverable support software create another problem; they are very difficult to establish.

There are situations in which configuration control should be exercised throughout the life-cycle maintenance of support software—it is when support software programs are installed at more than one location. Then, stringent configuration control must be exercised in order to avoid the problems associated

6. MIL-STD-1679 (Navy), Weapon System Software Development, 1 December 1978.

7. The size and complexity of the programs used for debugging, management control, configuration management, and document generation will vary. The program manager must therefore decide which of them are necessary to the modification and maintenance of the operational programs throughout the life cycle. Whether all these types of support software should be deliverable is still a subject of discussion.

with parallel modifications of physically separate programs. Operating system updates will also have great impact on support software performance. In addition, training of maintenance personnel who are responsible for keeping support software updated must be given consideration. The presence or lack of documentation will certainly impact the configuration management responsibility and control.

Advantages of Deliverable Support Tools

Some of the tools described above are valuable, especially assemblers, compilers, meta-assemblers and meta-compilers. These tools are necessary for the development of application software. With language processors, new application languages can be designed, and language translators can be developed in one-tenth of the time required to develop a translator from scratch. By the same token, these programs can be easily modified. At MDAC, a proprietary tool called "meta-translator" has been used over and over again to develop language processors, translators, and automated verification systems. The cost of application software development is reduced to a minimum. Syntax analyzers, for example, can now be written in a matter of weeks as opposed to man-years.

Because many of the new tools are designed to be portable (based on a standardized HOL or written in their own language), the cost of installation from one computer to another is minimal. Often it is a matter of investigating the system differences, providing appropriate adjustments and, perhaps, writing a bootstrap program.

Support software is an important item in the overall software picture, but it is much more difficult to control than the actual embedded software. Why? Support software may be developed originally by the contractor using independent research and development or other funds; therefore, it is not deliverable under the contract. It can also be subject to other restrictions, most of which have already been mentioned. Support software developed under contract to the government can be controlled in the same manner as the embedded software.

Conclusion

I have given a cursory review of software development techniques and the tools associated with those techniques in order to provide the configuration manager with some ideas on how to do a better job. It is hoped that the configuration manager will now have a better affinity for software problems arising both during the development phases and during the subsequent life-cycle management. He should have a better idea of when configuration control should begin, and what to look for when he is required to prepare a software configuration management plan. He may look for interfaces between software configuration management and hardware configuration management but, above all, he should know that there are still many software problems yet to be solved. ||

Meeting the Evolving Micro Requirement

88

Jerry L. Raveling

Configuration management is no longer the staid and stodgy discipline it once was. Rather, it is a discipline that, much like the computer industry, is undergoing a period of major change and improvement. These changes and improvements are designed to increase the management and cost effectiveness of configuration management, and to enhance its application throughout the life cycle of a system.

By all indications, configuration management will play a significant role in future computer resources acquisition management, for it can provide the management visibility into the design, development, test, and operations/maintenance processes that is demanded by today's computer resource acquisition and operational management environment. This article addresses the application of configuration management to the evolving world of microprograms, microprocessors, and microcomputers, an application that only recently has been given a degree of attention by government and industry. It is intended to provide insight into the current micro configuration management environment as viewed by a representative of industry.

Configuration Management Overview

Configuration management is the formal process applied to system and equipment programs for the identification, control, and accounting of the configuration of the system and all related equipment. It is also a means whereby design, engineering, and cost trade-off decisions are recorded, communicated, and controlled. The principles and procedures of configuration management have been developed and applied in many system programs during the past 25 years. They originated as a set of techniques for controlling and verifying changes to operational military equipment. These were revised and expanded in the early 1960s to cover the preparation and control of specifications during the definition and acquisition phases of a system program. Application of these procedures to computer software was begun in the mid-1960s, a process that only in recent years has shown signs of maturity. Now we are again expanding the discipline to meet the evolving configuration management requirements of microprograms, microprocessors, and microcomputers.

Why do we need configuration management? Because configuration management is truly a discipline that is designed and required for total life cycle management. When properly implemented, configuration management has provided

Jerry L. Raveling is a Staff Consultant on future product planning for Sperry Univac Defense Systems Division. He has over 16 years' experience in the management, development, and control of medium- to large-scale military data processing systems, with emphasis on software configuration management and quality assurance. He is the chairman of the 1979 EIA G-33 Configuration and Data Management Workshop. Mr. Raveling holds a B.S. degree in education from the University of Colorado.

significant improvements in overall management visibility, and has supported the cost-effective development and maintenance of systems and their related hardware and software items. At the same time, it has been identified as one of the management disciplines that, if not applied judiciously, will result in ineffective expenditure of program funds, and increased life cycle costs.

The basic framework for configuration management is the system life cycle process (Figure 1) with its associated baselines, specifications, design reviews, and audits. As we progress through the life cycle, configuration management is first applied only to specifications; then it is expanded to include not only the specifications, but other system documentation such as test plans/procedures. Finally, it is extended to the product itself—be it an aircraft, ship, tank, or an embedded computer resource (including the mini- and microcomputers, microprocessors, and associated computer programs or microprograms).

The major elements of configuration management are:

- Configuration identification;
- Configuration change control;
- Configuration status accounting;
- Configuration verification/audit.

Configuration identification is an evolutionary process. As we progress through the system life cycle, the technical documentation (specifications, drawings, and lists) required to describe the functional and physical characteristics of an item are identified and baselined. Configuration identification ensures that the documentation and products are current, approved, and available for use as required.

Configuration change control is the most apparent and formal aspect of configuration management. It provides for the review of proposed system changes for hardware, software, and firmware; classification of changes; resource management; and change tracking and traceability.

Configuration status accounting provides the means by which actions affecting the configuration of a system configuration item are recorded and reported to project management.

Configuration verification is the audit portion of configuration management, wherein we verify compliance with specifications and other contract configuration management requirements, and ensure proper change implementation and release.

Configuration management provides the means to:

- Define the impact on design function, performance, and cost of a change to the system hardware, software, and firmware;
- Promptly evaluate proposed changes;
- Incorporate changes in a timely manner;

- Ensure that adequate requirements and support documentation are developed and maintained;
- Define the configuration of the system;
- Provide for uniformity and standardization in the development process.

Micro Configuration Management

Now, given this basic framework, what effect does the developing world of microprograms, microprocessors, and microcomputers have on configuration management? Let me first state that the application of configuration management principles and procedures to the micro family is in its early formative stages; few of the formal government configuration management standards recognize or address, even in general terms, this application.

What I am about to describe is not a cookbook approach to configuration management; rather, it is intended to provide you with some awareness of the complexity of the subject and point out the need for careful planning and coordination of future configuration management policies and procedures for microprograms, microprocessors, and microcomputers.

Figure 2 depicts what is, perhaps, a typical micro configuration identification scheme. First, the basic requirements for the firmware and microprocessor are identified in a hardware specification. Next, the microprogram is identified by its listing and the media upon which the program is stored. If, as is often the case, the program must be partitioned onto two or more chips, a truth table and burn tape will be developed for each chip. The raw programmable read-only memory (PROM) device must then be identified, and it, in turn, must be reidentified once the program data has been burned in. The devices are then associated with their parent circuit card assembly and, finally, with a major hardware component. This type of configuration identification scheme has proven adequate for most hardware-intensive application; that is, where the chip(s) or microprocessor has been used to replace what has traditionally been accomplished through hard-wired circuitry.

Management concern currently is in the migration to microprocessors of what heretofore have been traditional software applications. We are now finding that the identification scheme shown in Figure 2 is no longer totally adequate, as it does not provide for positive control of the related software media and documentation.

In recognition of this deficiency, the Air Force Systems Command (AFSC) recently issued Supplement 1 to AFR 800-14, the USAF's bible on management of computer resources in systems. The supplement establishes the basis for Air Force configuration management of microprocessors, microprograms, and microcomputers. It states that microprocessors and microcomputers are to be identified and

controlled as hardware items, while microprograms, or firmware, are to be managed as computer program configuration items (CPCIs) or as components thereof. Similar policy statements have also been issued recently by the U.S. Navy and the U.S. Army (reference MIL-STD-1679, NAVELEXINST 5200.23, and Draft MIL-S-52779).

While not wishing to minimize the impact of this policy direction on hardware configuration identification practices, I believe the greatest changes in configuration management practices are to be found in the microprogram area. Why? Because stating that microprograms/firmware shall be managed as CPCIs or computer program components (CPCs) creates a dramatic expansion in the configuration identification requirements, since now, by definition, the microprogram/firmware becomes:

- An essential system element that must be a visible portion of the overall system design;
- An element that must be managed in accordance with DOD's joint service/agency configuration management regulation. This regulation directs the use of the MIL-STDs or specifications that establish configuration identification, control, status accounting, and verification requirements;
- As an essential system element, the design and cost factors of the microprogram must be addressed early in the system's life cycle, along with those of the major hardware and software elements;
- And, finally, configuration management is to be controlled by a single authority. No longer can hardware, software, and firmware configuration management exist as separate entities. A systems approach to configuration management is being directed on both the procurement command and the contractor.

Let us now examine the specific application of these policies. Configuration identification is a good place to start. We now need to identify our microprogram (see Figure 3):

- In a performance specification, either a B5 computer program development specification, as found in MIL-STD-483/490, or a program performance specification (PPS), as described in SECNAVINST 3560.1, or the recently issued MIL-STD-1679;
- In a product specification, either the C5, as found in MIL-STD-483/490, or the program design specification (PDS) and program description document (PDD) as described in SECNAVINST 3560.1 and MIL-STD-1679;
- In an interface design specification (IDS) to describe the microprogram/microprocessor interfaces (HW/HW, SW/HW, FW/FW, etc.);
- By preparing test plan and test procedures to describe how, when, and where the item(s) shall be qualified;

- Identifying the deliverable program media itself, to include the master program and its listings. If partitioning is required, then these program breakouts must also be identified, as well as all required burn tapes needed to prepare programmable read-only memory devices.

Further, DOD Directive 5000.29 espouses the need for including support software—the software used to produce and test deliverable software, or microprograms—as a deliverable item in a procurement. If this requirement is levied in a contract, then the list of required identification items expands significantly to include another set of documents, products, and test data. When these requirements are combined with the hardware identification requirements, the configuration identification problem virtually explodes.

Perhaps the extensive amount of documentation, program media, and hardware components that must be identified for configuration management purposes would not be quite so imposing, were it not for one major consideration: What if the microprogram has an error? What if the chip has a fault? What if the design requirements have changed? Now the policies and procedures of configuration control come into full effect. But wait. Is it a software error? Is it a hardware error? Or is it both? A system of configuration control must be established to provide for:

- A way of reporting the error to the proper engineering/management authorities, and for requesting that the actual change be developed;
- Change criteria, as well as a method of classification of the change;
- A means to analyze, approve, implement, and monitor the change throughout the process, which normally entails the establishment of a Configuration Control Board (CCB), Interface Control Working Group (ICWG), and a Configuration Management Office (CMO).

If the error is attributed to the hardware production process, it may involve only replacing the errant chip or microprocessor with a new one; replacing or resoldering a connector; or perhaps replacing the entire circuit card assembly (see Figure 4). If a design change is required or a design error has occurred, then specifications, drawings, test documents, and the components themselves must be changed. If, however, it is a software error, then one must trace back through the hardware identification maze and enter the world of software. If it is an implementation or production error, the code, master tape, master list, and the affected truth tables and PROMs must be modified. Changes will also impact the microprogram documentation and test data, and may also impact the related support software, its test media, and documentation.

Micro Configuration Management Considerations

Implementation of configuration management for micros shall have, as has

been described, a significant effect across the configuration management discipline. This section will provide you with some random "snapshot" thoughts of what these effects will be, and will also provide some further ideas on how configuration management might be applied. These thoughts should be considered as preliminary in nature and, thus, subject to revision as further experience is gained.

CONTROLLED RELEASE

First, the release of microprograms and microprocessors must be controlled through configuration management. Basic micro configuration management policy should address the following areas:

- Microprograms/firmware must be documented and identified.
- Once a PROM is programmed, the PROM part number must be changed.
- When a microprogram, which resides in a PROM, is modified, then both the microprogram identification and the PROM identification must be changed.
- Erasable PROM (EPROM) part numbers need not change if the microprogram residing in the EPROM is changed; however,
- EPROM resident program identification must change whenever the program is modified.
- Support software used to develop a microprogram/firmware must be identified and controlled.

Are there times when you will not wish (or be required) to maintain configuration control over a micro product? Let's examine this question. If a microprogram or microprocessor is to be released for formal use, then it must, I believe, be subject to configuration control. Even a throw-away part needs control, since it must, under normal circumstances, be replaced by another. The microprogram, while not modifiable/maintainable in the normal software sense, must also be controlled so as to provide the capability for error correction, enhancement, or re-release.

TAILOR REQUIREMENTS

When configuration management has proven ineffective in the past, it has usually been the result of its being applied too early in the development process, or as a result of inflexible documentation requirements. This ineffectiveness can be prevented by carefully tailoring the configuration management and development process to avoid premature control or unnecessarily restrictive documentation requirements. Examples of the tailoring process for micro configuration management are:

- Use of commercial type specifications in lieu of formal government specifications;
- Adoption of a non-complex specification format, and its inclusion within

MIL-STD-483, 490, and 1679. A draft of this type of specification was issued by AFSC last year. It is basically a development/performance specification that includes the program listing to describe the detailed design.

- Definition of the type of micro-application intended; that is, if it is a hardware-intensive application, then perhaps the firmware could be documented as an appendix to the hardware specification, while software-intensive applications would be documented as a computer program;
- Similar accommodations could be made in the test plan/test procedure area.

INTERNAL/FORMAL CONFIGURATION CONTROL

A system of internal (internal to the developing contractor) and formal (customer) configuration control is also encouraged. Internal control can be designed so that it is less stringent, yet effective when the stage of specification or product development is considered. Control of independent research and development product development is normally not required, nor advisable, unless the resultant product will be used in a production-type application.

STATUS ACCOUNTING

Given the extensive configuration identification and control tasks required by micros as described in the preceding section, the status accounting function is dramatically affected by the significant increase in the number of identified items for the microprogram/microprocessor control. It must provide status information on all affected items once they are baselined. The need to automate the status accounting process, if not already accomplished, becomes very evident.

AUDITS

The audit process is also impacted significantly by micros, since the documentation of the microprogram/microprocessor crosses the software/hardware boundary and, more importantly, qualification test results may be found in both software and hardware tests, depending on the microprocessor application.

TEST AND QUALITY ASSURANCE

New consideration must be given to the interface between configuration management and the test, verification and validation (V&V), and quality assurance (QA) groups. When do you place the micro code under control? When do you start internal test? As software, before the burn-in process, or after burn-in? I have no all-encompassing answer; again, a tailored program of management and control is advised.

LIBRARY CONTROLS

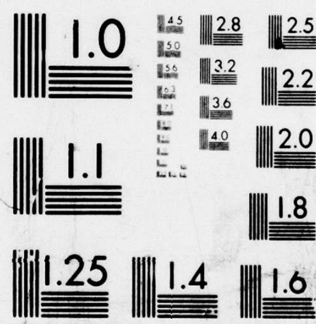
Recently, much has been written and discussed about the need for a soft-

AD-A074 518 DEFENSE SYSTEMS MANAGEMENT COLL FORT BELVOIR VA F/G 5/2
DEFENSE SYSTEMS MANAGEMENT REVIEW. VOLUME 2, NUMBER 3. SUMMER 1--ETC(U)
1979

UNCLASSIFIED NL

2 OF 2
AD
A074518

END
DATE
FILMED
11-79
DDC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

ware/firmware control library to work in conjunction with configuration management. Libraries offer positive control of developing microprograms and related test and support software. With the probable requirement to maintain configuration control over microprograms for long periods of time (estimates range from 5 to 20 years), a library offers the developer a method of providing positive control and change traceability, as well as enhancing the modification or enhancement process.

DOD and Industry Initiatives

What is DOD doing to make industry's jobs easier? Many of the current configuration management standards were prepared in the 1960s; hence, the documents are primarily oriented to hardware configuration management. Changes to these documents are required to reflect new DOD policy on the acquisition of systems; to reflect software configuration management practices; and, now, to reflect microprocessor policy and practices.

The Department of Defense has embarked on a configuration management standardization program that should have a positive effect on this general problem area. The DOD standards are now being examined and are scheduled to be revised, eliminated, or combined in order to:

- Provide guidance on tailoring of requirements;
- Eliminate overlapping and conflicting requirements;
- Fill voids in those requirements not currently addressed;
- Eliminate overly restrictive practices;
- Increase configuration management knowledge;
- Improve cross-discipline integration·
- Standardize terminology;
- Generally improve configuration management.

Examples of this process are:

- A new DOD Directive 5010.19 on configuration management replaces the old 5010.19 and 5010.21.
- MIL-STD-480A is scheduled for revision to incorporate software provisions.
- Revisions to MIL-STD-483 and 490 are planned.
- A new MIL-STD-XXX on CM is planned.
- The Navy recently released MIL-STD-1679, a document that places increased emphasis on structured software development.
- The USAF has published a software acquisition management (SAM) guidebook series, and the Navy has embarked on a similar program.

Conclusion

As I stated at the outset, I cannot offer a "cookbook" approach to

microprogram, microprocessor, and microcomputer configuration management. I do hope I have placed the application of configuration management to the world of micros in a clearer light; have pointed out the absence of written standards/guidelines on the application of configuration management; that this article has shown that existing hardware/software configuration management techniques do have direct/indirect application to micro configuration management; and finally, that we all realize the need to continue to explore, and to develop improved configuration management policies and practices.

97

The Impact of Today's Electronics Technology on Systems Acquisition

Lieutenant General Robert T. Marsh, USAF

It does not require an especially close look at the Air Force today to notice our great—and growing—dependence on electronics. As our forces have declined in numbers over the past decade, we have looked to electronics to help us redress the military balance through increased effectiveness of the forces we do have. But the influence of electronics goes well beyond the substitution of quality for quantity, and beyond "force multiplier" considerations. Rapidly advancing technology has meant totally new possibilities and capabilities that profoundly affect not only the way we develop our systems for the future, but also our basic doctrine and our thinking about the way aerospace forces will be employed.

One of the more significant trends in the Air Force in recent years has been the emerging emphasis on C^3—command, control, and communications—and it is in this area of C^3 that our dependence on electronics is most intense. The acquisition of C^3 systems is, of course, our primary mission at Electronic Systems Division, which leads me to the focus of this paper..

This has been called "the age of computational plenty"—a vast leap ahead in computer technology, accompanied by a decline in computational costs—all this largely brought on by microprocessor/large scale and very large scale integration. Looking ahead in trends relative to C^3, we envision that processing power and cost effectiveness will each continue to increase by at least one order of magnitude per decade, while memory costs decrease an order of magnitude per decade. Microprocessors will have major impact in the areas of sensors, signal processing, surveillance, and communications processing. Distributed data processing will become more the rule than the exception, and substantial R&D will be needed in distributed data base management.

Although this boom has been a blessing, the systems development and acquisition management tasks for C^3 have become more complicated, particularly for software. The multitude of design options now available, coupled with a proliferation of processing devices, demands standardization and control. These are the problems we are now facing and pursuing vigorously.

In many ways, our problem has shifted from one of technology itself to one of how to manage the technology and how to make the right choices.

We in the military are not alone in reaping the benefits of the new technology, nor are we the only ones with a job on our hands to assimilate and cope with it. Our situation, though, is not precisely the same as that of the commercial world.

Lieutenant General Robert T. Marsh is Commander, Electronic Systems Division, Air Force Systems Command. Previous assignments include duties as Project Officer, NAVAHO and MATADOR/MACE; Staff Officer and Executive Officer in the office of the Deputy Chief of Staff for Research and Development; Deputy for Reconnaissance, Strike and Electronic Warfare (AFSC); Deputy Chief of Staff for Development Plans and Systems (AFSC); and Vice Commander, Air Force Systems Command. Lt. Gen. Marsh holds degrees from the U.S. Military Academy and the University of Michigan.

Our decision-making is driven by the very nature of the military mission. There is a tendency toward "full speed ahead"..."the need is pressing, so let's get on with it." From a military viewpoint, that tendency isn't all wrong. Potential conflict may not wait until we are ready for it. If war comes, a *good* system in the field is worth any number of *ideal* systems on the drawing board. We must balance off time and cost against the advantages of technology, and against the improved performance that technology might give us. Our situation is also different because of the nature of the defense systems acquisition process with its lengthy cycle and its step-by-step decision-making. It poses the inherent risk of technological bypass. Finally, the life span of a military system is considerably longer than that of the average commercial system.

Making the Best Choice

The difficulties are obvious, I think, in trying to come up with systems that will not be obsolete by the time we get them into the field and, beyond that, be capable of growth to evolve with requirements and technology over the many years we expect to keep them in use. One dimension of impact of the current accelerating electronic technology, then, has to do with making the best choice among the choices available. The pressure is always on to pick the latest technology. This strategy is encouraged by the natural desire not to be outdated before we get the system built. Technology is improving so rapidly that the typical advanced component stays out in front for less than 5 years. Even if we pick the latest, whatever is at the leading edge today, we may have an obsolescence problem by the end of the acquisition cycle.

In the early 70s, the PMOS 4-bit microprocessor revolutionized the semiconductor industry. Now microprocessors have grown to 16 bits and use NMOS technology. Those military systems that committed to the use of the 4-bit PMOS microprocessors in the design stage must worry about the availability of replacement parts, the poor performance of their components compared to what could now be acquired, and the questionable cost effectiveness of redesigns and retrofits. Choosing the latest is also the high-risk option, so there is a corresponding pressure to go with the tried and true.

Either way, we face the additional question of whether our choice will remain available and supportable. The marketplace is essentially geared to commercial demand, and we could easily find ourselves stuck with an item that has been discontinued, either because a different product that's a real driver comes along, or because the item we picked is not a commercial success.

We need to explore means for protecting ourselves against diminishing sources of supply. During the past few months, five manufacturers have told us they intend to terminate production of selected electronic devices. In one case, we are buying over 400,000 copies of a series of integrated circuits that will be

discontinued. This number will be needed to support future production and spares requirements—but should an additional need for still more arise later on, we would have no alternative but to redesign the affected portions of the system. If King Solomon were alive today, we'd love to have him with us in Systems Command, helping us to make the wise choices among available technologies.

Designing for Change

Another impact of rapidly advancing technology is on our philosophy of packaging and supporting our systems. We all recognize the importance of being able to change missions or interfaces without costly retrofits of existing systems. Fortunately, modern electronic trends allow circuitry to be altered without substantial changes in hardware. The bit-slice microprocessor, for example, can have even its instruction set altered using new software through microprogramming techniques. Unfortunately, there are often large software costs involved. There is also a lack of software standards, which increases the difficulty in making rapid changes.

It would be ideal if we could design-in all the growth flexibility at the board and chip level, so we could pull them out and replace them as new technology comes along—and do it without disrupting the rest of the system. But we aren't that smart yet.

Consequently, we have to attack the problem at a different level. We must establish a standard, not in design of discrete modules, but in the design of the way they interconnect. At ESD, we are working toward an electronic bus-type standard for C^3 systems and centers. This would mean that anyone designing a new element—a new computer, peripheral, or whatever—would adhere to the standard and build interoperability into the element being added on.

The new element could internally exploit technological advance while plugging into the central bus just as its predecessor did, thus enabling evolutionary change to occur. Furthermore, this would help to solve some of the interoperability problems that have plagued C^3 systems for a long, long time. Said another way: As we think form, fit and function, we at ESD are thinking less about form, and more about fit and function.

Today's fast-moving electronics technology is also changing our concepts of reliability and maintainability.

With some of our systems...space systems, for example, or the unattended SEEK FROST radars that ESD is developing for the Distant Early Warning Line...we must have an extreme degree of reliability. It's not a matter of "fail and repair." These systems must be built so as not to fail, because we do not have ready access to them. The highest reliability is also necessary for certain critical strategic weapon systems. While this high degree of reliability is most vital in key

or remote systems, we are, in a sense, coming to where we will have to treat all of our systems as if they were unattended, even if we have ready access to them.

This is partly because the operation itself requires that degree of reliability—but it's also because the systems will be too complex for us to legitimately expect a serviceman in the field to be able to service and repair them. This pretty well precludes preventive maintenance in the field. In some cases, repair may not be possible. You can't repair an LSI chip. Even where repair is possible, it may not be cost effective. We seem to be headed toward a support concept of "discard on failure" rather than the old concept of repair.

The search for reliability again puts us at a fork in the road. One way is the leading edge of technology. The other way is the tried and true. All it takes is one unreliable component to doom a satellite or a remote system, where maintenance is difficult or impossible. By the time confidence in a given part has been established—usually at great expense—that part is obsolete. We still use germanium PNP transistors in critical strategic weapon systems, not because this 20-year-old technology is so reliable, but because we know exactly what its reliability it.

On the other hand, the degree of reliability we seek is inextricably linked with the new, galloping technology, and most often, we'll have to take that fork of the road. The increases that are possible in chip density allow the designer to pack in more redundancy, and improve reliability that way.

The integration of digital logic and analog circuitry leads to fewer external interconnections, where a bad contact could mean unreliable operation. In many instances we are now able to have very large memory systems—either bubble memories or charge-coupled diode memories—without the mechanical unreliability of discs, drums, or tapes. Better fabrication and manufacturing techniques leave us less subject to the variable of human error.

Built-in Testing

This greater need for reliability and less reliance on repair, especially in the field, points clearly toward more built-in testing, and more self-healing features in our equipment for the future, which brings us to the all-important issue of testing. There was a time when every active element in a given electronic subsystem could be thoroughly tested individually. This is no longer the case, as tens of thousands of elements are integrated onto tiny chips of silicon. The cost and reliability advantages of using these highly integrated circuits are well known.

But a monumental testing problem has been created. The number of permutations of test conditions possible for a single circuit are astronomically large. Design errors in some popular circuits have gone undetected for years—until they appeared in routine use. One can easily imagine the dismay of a pilot who suddenly finds that his bomb load won't release because of a design error—that being

an actual example. In addition to design errors, random defects are bound to occur in the fabrication of highly integrated chips. These defects could produce the same effect as an initial design error, but they are much harder and more costly to find.

At present, we are having a real headache with our built-in-test/fault isolation sub-systems. Some systems now in the field have been experiencing false alarm rates as high as 30 percent, and fault isolation effectiveness as low as 50 percent. Such deficiencies often prevent mission completion or cause large quantities of "black boxes" to be erroneously removed and shipped to a depot for repair. All this has resulted in large numbers of costly spares tied up in the supply pipeline.

A tough job lies ahead to make our testing better. You can't completely exercise a chip—put it through all of the tasks it may be called upon to perform—so it's going to take a lot of clever design work to determine a set of samples that are sufficiently indicative of how a piece of equipment will function.

And as we rely increasingly on off-line testing and on built-in testing, our results absolutely must get better. As we see it, the great growth in logic design and engineering effort in the near future is going to be in a design of tests.

There is a very interesting technology that ESD's Rome Air Development Center has come up with. Most solid-state circuits fail not from normal wear and use, but from faults in design, fabrication, or manufacture. The new Rome technology makes non-destructive observation and analysis of microcircuits possible. It involves coating the microcircuit with liquid crystal, and slowing down its operation so that the progression of logical steps can be watched as the circuit is put through various test functions. This is not a production technique—it's an analysis technique. But it holds promise for use in the initial design phase of a system, where we will be able to see where potential failures might be.

Cost Versus Performance

In terms of cost/performance trade-offs, cost considerations usually win out over performance, although modern electronics are certainly faster, less power-consuming, and more capable than the older technologies. The Air Force would like to reap the full potential of modern technologies, but this usually involves developing highly specialized components that have no commercial base. Digital logic functions are usually capable of operating only up to a few megahertz. This data transfer rate severely limits our ability to manipulate the data. Technologies are rapidly being improved, however, and full base-band on-board data processing is expected to soon become a reality.

The bottom line to the overall impact of any development is life-cycle cost. Initial acquisition costs for electronics have been bucking the inflation trend and dropping steadily for years. The big news is that life-cycle electronics costs are now dropping because of more reliable and maintenance-free operation. If we continue to develop adequate hardware and software standards, many of the

problems I've been talking about will be solved, and the bottom line costs significantly improved.

A few other observations: We in the military do not have all of the answers. A great body of electronic technological knowledge lies in the commercial and industrial world. Now, more than ever, it behooves us to draw upon that body of knowledge, to state our need, and then let the designer design the system.

A recent example from our COMBAT GRANDE program illustrates. We had asked for the standard flow charts and narrative to define the computer systems being developed for COMBAT GRANDE. They were a pain in the neck to maintain, and they were seldom timely. The contractor suggested that instead of the "flows and prose," we accept a more contemporary product, hierarchical input process output (HIPO) diagrams. We did, and it proved to be a significant aid in developing the system. We're thankful that the contractor offered us this improvement rather than what we asked for.

Lesson Learned and Relearned

Over the years, we have learned and relearned the lesson that we ought to start with functional requirements and not try to state the conclusion in the same breath as we phrase the question. Now we have formal direction to help us remember this lesson learned. OMB Circular A-109 and the directives that embody its concepts require us to make our Mission Element Need Statement (MENS) in operational terms—the problem, or what we need to be able to do—rather than in terms of equipment. This way of doing business had many pluses, one of them being that it can help us adjust to and take best advantage of new technology as it comes along. We will not be blindly bound to precedent. And since we must begin with proven needs, we will not be pursuing technology just because it is there. Instead, it forces us to seek the best match between specific requirements and available technology. Circular A-109 also forces us to put more emphasis on "front-end" thinking—the early phases of the acquisition process—so we will do a better job of definition and of analyzing the life-cycle aspects of acquisition that technology and other factors have made so important.

The topics discussed here do not by any means exhaust the list of ways that today's galloping electronic technology is affecting our systems acquisition. That impact pervades almost everything we do. It has already helped us to solve some of our problems. It will, in the near future, help us solve still more. It has given us dramatically new capabilities.

At the same time, the rate of technological progress has presented us with some formidable assimilation and management challenges, not all of which do we have an adequate handle on yet. The real challenge to those of us in the acquisition business is to employ all the technical and management expertise at our disposal so that we not only cope with galloping electronic technology, but also reap great—perhaps yet untold—advantages from it. ‖

Background of Study on Specifications and Standards

103

Dr. Joseph F. Shea

The Defense Science Board Task Force study on specifications and standards was initiated to investigate the "unreasonable costs arising from Military Specifications and Standards." As a group, we approached our subject with a bias against the existing body of some 40,000 specifications and standards contained in the DODISS (Department of Defense Index of Specifications and Standards): "There are too many, the requirements are unnecessarily severe, some are obsolete, they dictate how to do something, not what is required...," in other words, the familiar litany of industry's complaints about excessively detailed customer documentation.

Presented with a charter to challenge the existing culture, we began by examining some 20 flagrant examples, identified by industry, of excessive cost arising from particular specifications. In almost every case, reading the document revealed that it contained much more flexibility than appears to be used in practice. The "excessive cost" resulted from a failure to utilize this flexibility in a reasonable way, rather than from a fundamental problem with the specification itself. Interestingly, industry was as guilty of overreacting in interpretation of the requirements as government was of overenforcement.

We began to realize that the problem was not exactly as we had originally conceived it. Our bias began to change.

Specifications and standards improve the quality of a product by defining proven components, fabrication techniques, test procedures, and management disciplines, while at the same time reducing development risk and lowering production cost. Specifications are essential to technical procurement, and all responsible organizations, whether involved in commercial or government products, invoke them to a considerable degree.

Commercial specifications address the needs of a particular corporation or industry. There is a body of opinion that would replace MIL SPECS with whatever commercial counterparts exist. Discussions with the organizations responsible for national industry standards, such as the American Society for Testing Materials

Editor's Note: The Task Force on Specifications and Standards Improvement was chartered as a panel of the Defense Science Board in 1974 under the chairmanship of Dr. J. F. Shea. The Task Force completed its deliberations and published a final report in April 1977. This article is a summary of the background, findings, and recommendations of that study as presented in that final report.

Dr. Joseph F. Shea is a Senior Vice President of Raytheon Company and Special Assistant to its President. Before joining Raytheon in 1968, Dr. Shea was Vice President of Polaroid Corporation responsible for product design. From 1962 to 1967 he served with NASA as Deputy Director of Manned Space Flight and as Manager of the Apollo Spacecraft Program. He has also served as Manager of Advanced Research and Development for General Motors' AC Electronics Division and as a Research Mathematician with Bell Telephone Laboratories. Dr. Shea holds a B.S. degree in engineering mechanics, and M.S. and Ph.D. degrees from the University of Michigan.

(ASTM), indicated that, in general, MIL SPECS are considered technically superior to their commercial counterparts, which often had to accommodate the least common denominator of performance in an industry. Procurement specifications from large commercial companies encompass the same level of requirements found in government documents. Although the details differ, to a supplier they tend to look almost the same.

Two fundamental aspects of specifications became apparent:

- They are essential.
- Considering the breadth of today's technology, no standard can be unique. Selection of a standard is, therefore, an arbitrary decision.

DOD, in developing specifications, must take into account procurement regulations that encompass a wide range of potential suppliers, from established companies with proven track records to small businesses trying to get a start. Specifications and standards can enable inexperienced companies to learn how to produce acceptable products while providing the procuring agency with the leverage to ensure that suppliers use materials and processes that will produce a quality product.

Why Are Specifications Maligned?

If one accepts the need for specifications, then a question arises as to why they are so maligned. The answer is that, to be effective, a specification must be an essentially arbitrary selection of one or more proven ways to accomplish a goal from a much larger sub-set of possible approaches. There is no one unique way, for example, to solder correctly. But to guard against the many improper possibilities, a soldering specification will require a specific, proven approach, thereby ruling out other equally acceptable alternatives.

The same situation obtains in the choice of standard parts, text specifications, or assurance plans. Alternate acceptable choices exist across the entire spectrum encompassed by the Defense Standardization Program. The essence of standardization is making reasonable economic, flexible selections of the standards to be promulgated, and the acceptance of those choices by both government and industry users.

This is much more easily said than done, because personal convictions, experience, even economic survival, can and do enter into the judgment of which choice is correct. Ideally, individuals and organizations involved in standardization should recognize that the inherent gains can be achieved only by complying with those choices, not by continuing to advocate equivalent or marginally improved standards; that is, accept the arbitrary nature of specifications and standards in today's technological world.

The Task Force's review convinced us that the present body of specifications

and standards is reasonably good. As noted earlier, most of the instances of "excessive cost" we examined resulted from a failure to utilize in a reasonable way the flexibility, or options, incorporated in the specification, rather than from a fundamental problem with the specification itself. In general, the specifications contain much more latitude than appears to be used in practice.

These observations can be reconciled with the generally accepted view of specifications by observing that, in the mass of some 40,000 documents contained in the Department of Defense Index of Specifications and Standards, there are bound to be some ludicrous requirements that make great anecdotes—a 15-page specification for chewing gum comes to mind. There is a tendency to use such documents to disparage the system in general, rather than to look for its strengths.

A Two-Step Solution

This is not to say that the system doesn't need improvement. There is much that can be done. The Task Force concluded that, while the present body of specifications and standards is "adequate" to current needs, DOD does not employ a coherent philosophy for the development, revision, or administration of specifications. Excess costs are associated with the specifications, but primarily in the way they are used. The Task Force recommends that solution of these problems be achieved in two steps:

- By an immediate program throughout the Department of Defense and industry to improve the climate of contractual application; and
- By an evolutionary program to improve the existing body of specifications.

The first step must be a joint government/industry effort to effectively tailor the application of specifications and standards. The second step is primarily a government responsibility, supported by competent, interested industry groups.

At the risk of being somewhat redundant, let me summarize our major findings:

- Specifications and standards are essential to technical procurement.
- The present body of military specifications and standards is adequate to the needs of the Department of Defense.
- Specifications and standards contain, for DOD, a corporate history of lessons learned. They communicate what and how to perform and thus restrict designers' options in an effort to reduce the government's risk and, in principle, to achieve lower cost.
- Specifications serve as a primer for the inexperienced as well as a safeguard to help assure quality products.
- Of the 40,000+ specifications and standards listed in the Department of Defense index, major cost impact can arise from the non-product variety

(i.e., general design requirements, documentation, management).

- Major payoff for improvement in specifications and standards will come initially in their method of application, followed by longer-range improvements in substantive content. In this connection:
  —Specifications contain tailorable alternatives which, in many cases, are ignored.
  —Excessive costs arise from misapplication, overapplication, premature application, and uncontrolled callouts of referenced documents.
  —Requirements for contractor demonstration of compliance can be excessive.
  —Excessive management systems and plans are required in non-system-related specifications.

Improving the Climate of Application

Improving the climate of application requires the use of common sense in the adoption, interpretation, and application of specifications.

As a starting point, we believe that industry can do much to make conformance a way of life. A large fraction of the cost associated with a specification, be it for soldering, standard parts, or management systems, arises from changes in normal procedures that may be required for compliance with a particular specified approach, or from the superimposition of a prescribed compliant system on an already existing structure. It would seem to be incumbent upon defense contractors to establish their design standards, processes, and program control systems to conform with MIL SPECS. Once this is done and the systems are used for both internal and external purposes, any incremental cost of compliance should virtually disappear.

But conforming doesn't mean overreacting. Many of the troublesome specifications leave wide latitude for interpretation. For example, DOD-D-1000, which concerns drawing requirements, contains several levels of applicability. The Task Force found that, very often, standard practices tend to go to the upper bound—the most expensive interpretation of each of the levels. Although this undoubtedly results in fewer contractual arguments, the practice is not in the best interest of either government or industry.

Specifications that treat quality control, configuration management, reliability, or other disciplines, require that a contractor achieve a desirable end result—a reliable product of good quality—by establishing and following a set of procedures intended to achieve the goal. But, too often, tests of conformance are in terms of procedural compliance—not goal achievement.

For example, MIL-Q-9858A, Quality Program Requirements, requires "the establishment of a quality program to assure compliance with the requirements of

the contract. The program and procedures...shall be developed by the contractor.... [It] shall be documented." In effect, the contractor is asked to define, document, and follow the procedures required to achieve a quality product.

Philosophically, it seems reasonable that a self-defined set of procedures, rigorously adhered to, might be required to produce a quality product economically. However, the system can become expensive if the procedures are made overly elaborate in order to impress the "government representative" who reviews the system, or if the discipline breaks down and problems result. The government handbook on evaluation of a contractor's quality program which, at 35 pages long, is almost four times the size of MIL-Q-9858A, states that the "quality program is subject to the disapproval of the Government Representative whenever the contractor's procedures do not accomplish their objective." The message is clear. Don't have problems. Unfortunately, problems do occur. Too often the reaction is to add procedures rather than to get at the root cause. The result can be form without substance, effort without result or purpose, which causes a slackening of discipline that can cause further problems.

An improved climate of application does not mean that industry must blindly conform, or merely refine its interpretation of specifications. The MIL SPECS were written to cover a broad range of products destined for use in a myriad of operational requirements. They also tend to document the DOD corporate memory in an effort to avoid problems encountered on past programs. Inevitably, they contain redundant requirements or specific values that may be too extreme for a given case. This means that such specifications must be invoked and administered with common sense.

Tailoring

The process of using common sense in the application of specifications and standards is called "tailoring." In essence, this means using the specifications as a reasonable starting point, but modifying their applicability to suit the circumstances of a given program. Perhaps a better definition would be: "stop treating the specs as sacred."

Typically, tailoring can include, but is not limited to:

- Modification of quantitative requirements (such as a temperature range or a vibration level);
- Selection of the appropriate level of requirements (such as type of drawings);
- Selection of only a limited number of requirements within a specification;
- Substitution of commercial or industrial specifications;
- Elimination of MIL-SPEC requirements not applicable to the specific program situation at hand;
- Control of referenced documents.

Tailoring is intended to encourage responsible people to understand the real requirement and be in a position to waive and/or change the specification. The climate should be one in which it is accepted that situations frequently occur in which waivers are actually good for a program, and should be encouraged.

The Task Force recommends that DOD policies encourage tailoring:

- Before the request for proposal (RFP) is issued.
- During the life of a program.

The relatively large number of specifications required on a contract makes it impractical to tailor each one before calling it out. Such a process would extend the definition/validation phase unnecessarily, creating an almost impossible burden for the already overloaded government program manager. However, the Task Force was able to identify specifications which, because of their wide usage and broad applicability, were prime candidates for misapplication and misinterpretation. As part of the RFP preparation, the government program office should tailor a sub-set of these cost-driver specifications, some 10 to 15, both to establish the climate for tailoring, and to benefit from the cost avoidance involved.

These cost-driver specifications are not a hard-and-fast set. The potential offenders vary with the service and the program. Typically, they include general specifications for materials, parts and processes, environmental and test specifications, documentation, management, and the "ilities" (reliability, availability, maintainability, producibility, etc.).

Tailoring should continue throughout the life of a program, from advanced development RFP preparation, through engineering development, production, and deployment. In essence, tailoring is an extension of the trade-off principles of design-to-life-cycle cost (i.e., useful performance for affordable cost) to levels of detail that are not usually challenged.

Good Judgment Required

Tailoring cannot be dictated by a set of hard-and-fast ground rules. It requires management and technical judgment on the part of both government and industry personnel. Because a decision to modify or waive provisions in specifications involves the distinct possibility of being wrong, the tailoring program must be strongly supported and publicized by management if it is to succeed. The existing procurement environment is basically conservative and encourages cautious conformance rather than forceful ingenuity. The government program manager and the functional organizations supporting him must be encouraged to realize that strict, parochial application of specifications and standards is neither required nor desired.

On a longer term basis, the Task Force recommended that the Defense Standardization Program focus on improving the existing body of specifications and

standards.

Although the existing specifications are adequate to DOD needs, there remains considerable room for improvement. There are too many specifications, which are often difficult to read and interpret. They do not contain clear statements of the problem being solved, and are rarely self-contained. Because the originator of the specification is frequently far removed from the user, both functionally and geographically, cost of application has not been a paramount concern.

The Defense Standardization Program calls for a review of each specification every 5 years to determine whether revision is necessary. This revision cycle is a natural focus for the improvement of the body of specifications and standards.

The Task Force recommends that all revisions to specifications, and all new specifications, be justified by a statement of intent, approved by Defense Standardization Program management, prior to initiation of effort. The goals of any new draft should be identified in order of priority, including:

- Expected impact on the cost of applying the specification;
- Increased flexibility through clarification or increased options;
- Upgrading for technical currency;
- Consolidation with existing related specifications, either within or across service lines;
- Use of, or consolidation with, existing industrial specifications and standards;
- Improved readability;
- Planning for coordination with industry.

Although the priorities may vary, it is important to identify the expected impact on cost of application in order to avoid excessive technical refinement. Industry coordination works reasonably well in most cases, but should be strengthened by the provision of an appropriate higher level of DOD management to resolve industry/preparing agency differences before a new or revised specification is issued.

Effective control of specification generation and revision can result, over the next 5 or so years, in improvement in the DODISS by:

- Reduction in the total number of specifications;
- Consolidation with industry or national standards;
- Lower cost of application.

Configuration and Data Management

The field of data and configuration management permeates defense procurement, and is essential to the production of controlled, reliable hardware that can be supported in operational use. It also impinges very intimately on a contractor's

day-to-day approach to doing his business, more so, perhaps, than any other area except cost and schedule control. For this reason, opportunities for tailoring the essential elements of accurate data generation, and rigorous control of configuration to an organization's standard practices, seem to me to be a particularly fertile field. Specifications that require reformatting of data, or levels of control inappropriate to the phase or size of a program, can only increase costs and divert resources that might be more productively used to improve the depth of design and test activity.

A good example lies in the increased use of computer-aided design and the natural flow of its output in digital format to computer-aided manufacturing. Not only are designs more rigorously checked out, but the change for human error is reduced in converting the engineering data to production use. Requiring such data to be transcribed to traditional design data formats would seem to negate the productivity gains from the new technology. In this area, not just tailoring, but fundamental specification revision appears appropriate.

One of the troubles with a concept such as tailoring is that, while it sounds good in principle, every application requires someone to make a decision. In the absence of a body of experience, the decision-maker may have trouble translating the general principles to concrete action. There is already a tendency to institutionalize tailoring—handbooks on how to do it, formal procedures, *et al.* The antidotes to this trend are examples of what makes sense and what, when identified by industry groups, forms the basis for future industry and government application.

One last point: When we adapt a specification to a particular situation, or waive provisions that are not germane, we are not short-changing the customer. Tailoring is not the process of providing as little as possible; rather, it is a route to optimizing what the customer gets for his money. Contracting organizations must be encouraged to forget the traditional approach that the deviations and waivers are indicative of lesser quality and performance and therefore require contractual consideration.

The Department of Defense is a large organization, and large organizations are slow to accept new ways of doing business. The concept of tailoring specifications has the potential for saving some percentage—say three to five—of defense acquisition costs. Although the percentage may seem small, the absolute dollars are significant. To achieve the gains, many individual decisions will be required, decisions that run counter to the prevailing culture of rigid conformance. We cannot expect such a change to take place overnight. Only a continued, dedicated effort by both defense and industry management over a number of years will produce the attitudes at the working levels to make the rational application of specifications and standards an accepted fact of procurement life. ‖

III Making Tailoring Work

Editor's Note: In late 1977 and early 1978, workshops on specification tailoring were conducted in San Diego, Calif., and Airlie, Va. The purpose of the workshops was to make government and industry managers at all levels aware of the need for cost-effective utilization of specifications and standards in materiel acquisition, and to make them also aware that such specifications and standards can be selectively applied to fit specific programs. This article is adapted from remarks presented by Carl Hershfield of GTE Sylvania at the San Diego Workshop, and by John Tormey of Rockwell International at the Airlie, Va., meeting.

DOD is placing priority efforts on tailoring as a result of the Defense Science Board study on improving specifications (as chaired by Dr. Joseph F. Shea) and DOD Directive 4120.21 (Specifications and Standards Application). The OMB Circular A-109 and DOD Directive 5000.1 also require tailoring of specifications and standards to the specific programs.

The term "tailoring" has been defined as the process by which the individual requirements (sections, paragraphs, or sentences) of program selected specifications and standards are evaluated to determine the extent to which each requirement is most suitable for a specific material acquisition phase, and the modification of these requirements, where necessary, to assure that each tailored document invoked states only the minimum needs.

It appears, based on the current documentation regarding tailoring, that we are embarking on an organized, priority effort by DOD that will enjoy top management attention in all of the defense agencies. In view of the increasing complexity and costs of systems, industry, through the industry associations, is also most interested in supporting efforts to achieve cost reductions through realistic engineering requirements.

The tailoring of specifications/standards and requests for proposal (RFPs) has as its objective eliminating unnecessary requirements so as to achieve realistic program requirements—all resulting in cost reduction.

We must recognize that there are many ways to structure a specification or standard to facilitate tailoring, and to do a good job we should use all the techniques available. Probably the one biggest mistake is the notion of reviewing all methods and then concentrating on only one. Good tailoring will consider all methods.

In restructuring specifications/standards to facilitate tailoring, it is suggested that the following be considered:

- Separating the essential or mandatory contractual paragraphs from the optional paragraphs;
- Identifying all requirements that can be specified in several levels related to specific application;

- Identifying paragraphs, requirements, and levels related to phases of the acquisition;
- Identifying paragraphs, requirements, and levels with type of contracts;
- Identifying requirements where lower levels may be proposed for cost reduction;
- Developing, where possible, a matrix of applicability for paragraphs or requirements against the programs, phases, criticality, importance, safety, cost, etc.

The requirement for the contractor to "tailor" should be specified in the RFP. The request for proposal is just that—a request for industry to propose a product or service to a buyer. To the Department of Defense the RFP is a most important document, just as it would be to any organization involved in the acquisition of major, complex products.

Without defending or supporting the existing RFP process, it is appropriate to acknowledge at the start that the RFP carries a sizeable burden in the role of communicating the requirements for a major defense system. We should expect to find a number of problems and issues that are the direct result of the complexity of the task and the pressures that come from numerous and diverse institutional interests involved in the action.

RFP Must Communicate

The RFP is generally DOD's first formal communication with industry on a competitive procurement. It is their opportunity to tell industry just why they are willing to spend some dollars, what's really important in the project (and thereby what isn't), what the procurement philosophy will be, and how they intend to grade industry proposals. Both DOD and industry have shown in recent years a consummate lack of skill in exploiting this communications opportunity. The stereotyped RFP format, which attempted to incorporate all the Armed Services Procurement Regulation (ASPR) instructions, evolved over the years to the point where it often managed in 250 pages or so to cloud issues with the skill of a magician. Whether this is a result of too liberal interpretation, or lack of understanding of basic program objectives by the writers, or just plain poor writing—or all of these—is not the point. The RFP must communicate to the prospective contractor, in simple terms, what is wanted.

The average RFP references various regulatory documents including the ASPR, DOD standards, military department specifications, etc. The mass of generalized, detailed, duplicated, and conflicting information in these documents may be overwhelming, and possibly only clearly understood by the author. Any effort on our part to simplify these documents would be a boon to contractors and the military services alike.

necessity to eliminate superfluous specification requirements. Industry strongly supports the report in this regard. High project costs benefit no one, either short range or long, but particularly long. However, profit is the industry driver and it shows up at the tailoring table. To be successful, the tailoring concept must accept the profit way of life; it must accept profit as one of the prime industrial motivations, and the government tailor must provide for the profit factor as a legitimate part of any specification negotiation. Contracting officers must stop seeking "consideration" for tailoring recommendations.

Intelligence is the *sine qua non* for the tailoring practitioner. In each and every tailoring session, technical facts, logic, costs, profit, schedule, performance, options, and trade-offs are put on the table by all the parties involved. Reason, objectivity, and above all, intelligence should dictate what, where, and how to tailor.

Industry Experience

What has been the industrial experience to date in tailoring during the precontract award periods? Is the new doctrine working? Has the Shea report and its supporting directions, advisories, guidelines, and the ASPR change cut away the bindings of cautious tradition? In a word, are we tailoring? The answer is an unqualified yes! Throughout those segments of aerospace in which I either move or can readily investigate, tailoring is the "in" word, a new enthusiasm, and its effects are usually good, frequently excellent, and occasionally spectacular. Program, project, and contract people have consistently praised the new look and are adopting it enthusiastically. At this early stage, however, all is not wine and roses. Some reports from the field are:

- The act of tailoring itself takes more time and money than some people imagined.
- Occasionally, severe tailoring-down is neither in company nor program interests.
- Informal, undocumented tailoring is definitely a waste of everyone's time, and could even be risky. It must be done seriously, deliberately, and well, with results in writing and signed.
- An invitation to tailor, or an explicit refusal to tailor, by the government side on a particular project, or specification, would have saved a lot of needless work.
- Uncertainty that all parties are sympathetic to tailored specifications (AF-PRO, DCAS, Source Selection Board, top-level government staff and functional units, etc.).
- Tailoring negotiations have generally tended to be less than the one-on-one exchange between equals, seeking a common solution. Rather we are

We cannot rewrite these regulations, but we can perform a valuable service that will streamline the responding proposals, reduce the volume of data, and possibly reduce contract costs. We must specify the degree to which each referenced document is applicable in the contract and the degree to which it is to be discussed in the proposal.

The yardstick for deciding on essential or nonessential data requirements can be simply stated: Is this information or requirement essential to the desired system and to program success, and will the data or the requirement actually serve these ends?

Five Simple Terms

The case for tailoring, both advocacy and non-advocacy, comes down to five simple terms: cost, personal security, *competition, profit,* and *intelligence.* These seem to underlie the entire subject. Because they come up again and again in discussions, let us look at them a moment.

Cost savings are promised to the extent that a tailored-down specification eliminates the need for the expenditure of labor to perform previously specified contract tasks. But, cost growth is feared to the extent that the amended specifications produced by tailoring have themselves required costly labor to produce and have engendered costly program uncertainties. In certain contracts involving one-shot-to-success systems, a tailored specification with a skipped section can be the door to single point failure and total project loss. One $1000 saved in an eliminated specification step may lose the one $20-million vehicle the program produced.

Personal security is the natural state that all parties to the program seek to achieve. From the system program office to the program manager, on through the rank and file of a project, everyone loves security. Now, the tailoring way of life does not enhance security for any but the competent, or the strongest. These people thrive on change. The uninformed, the peripheral folks, and the non-program oriented, shrink from tailoring because it introduces an element of uncertainty into their lives.

Competition is the heart of American industry, a reality of our democratic, free-enterprise system. We thrive on it; the best ones grow on it. We fight for it when we are on top; lower our voices when we are at the bottom. If tailoring doesn't impact competition, it does touch it, generating a reaction. So long as the new policy of tailoring enhances competition, there will be a solid response from the customer's side of the table. But when tailoring loses that competitive flavor, when the better tailoring firm is *not* rewarded, when its ideas are tossed into a common pot, when tailoring leads to industrial leveling, then the industrial interest may recede and we'll be back in the specification Middle Ages again.

Profit is one of tailoring's results that receives attention from industry. The Shea report properly addresses costs, government budget limitations, and the witnessing tailored specifications handed over or down; or expensive documented exchanges, sometimes with an adversary flavor to the operation.

- Pre-contract-award tailoring is occasionally a sterile exercise because either the parties involved are not acquainted, or the product system has yet to reveal itself in operation sufficiently to allow specifications to be fitted to it.
- Some pre-contract tailoring has been neutralized by the fact that the eventual RFP has called out everything, in an *untailored* mode, in order to bring all bidders to a common baseline.
- Tailoring currently appears to be more person-oriented than cost-oriented.
- A heavily tailored set of specifications requires considerable follow-up, explanations, and adjustments, in both government and company staff operations, if harmony is to be restored.
- In one instance, under a tailoring situation, the agency in effect threw the whole book into the RFP, with no apparent attempt being made to sort or eliminate, leaving the prospective bidders with the entire job.
- A major problem is the exact costs associated with the top 120 specifications. Not only must costs of application be estimated, but reasonable estimates of potential programmatic impact due to absence of the specification must also be done.
- With a well-informed, confident, forthright program office, contractors appear to have little difficulty with tailoring proposals. Less secure, less experienced program staff are more inclined to stick to the tried and true.
- Not every pre-RFP contractor technical team knows the nuances of the top 120 non-product specifications. Without their specialty people (configuration management, quality, data management, etc.), early tailoring cannot go much beyond arm waving.
- It seems to be shaping up that the contractors will be playing the starter's role in tailoring.